AFRL-IF-RS-TR-2007-27
Final Technical Report
January 2007

ARCHITECTURAL VULNERABILITIES OF THIRD-GENERATION PORTABLE DEVICES

Syracuse University

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-IF-RS-TR-2007-27 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
ANDREW J. KARAM Work Unit Manager	WARREN H. DEBANY, Jr. Technical Advisor, Information Grid Division Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JAN 2007	Final	Jan 04 – Sep 06

4. TITLE AND SUBTITLE

ARCHITECTURAL VULNERABILITIES OF THIRD-GENERATION PORTABLE DEVICES

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-04-2-0058

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Shiu-Kai Chin

5d. PROJECT NUMBER
2311

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
02

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Syracuse University
113 Bowne Hall
Syracuse NY 13244-1200

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFGB
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2007-27

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 07-029

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The development of the Handheld evidence Recovery Operator (HERO) has created a new paradigm for extracting evidence from portable devices without modifying the device. We developed a handheld evidence recovery operator to address the challenges created by user passwords and PINs, and a standalone tool to copy the data located on the device to an external storage medium without modifying the device.

15. SUBJECT TERMS

Forensic, Vulnerability, VOIP, Cellular Phone

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Andrew J. Karam
U	U	U	UL	77	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

List of Figures

1.0 OBJECTIVES

The objectives of this Cooperative Agreement Award No. FA8750-04-2-0058 and the subsequent ECP covered three distinct activities:

1.1 Investigate the forensic potential of Voice over IP devices and communications.

1.2 Develop techniques and procedures for performing digital forensic analysis on portable digital devices such as pagers and personal data assistants.

2.0 SUMMARY OF WORK PROPOSED

2.1. Investigate the forensic potential of Voice-Over-IP technology

The advent of Internet telephony, commonly known as voice over IP, created new challenges for law enforcement, and hindered them in the task of collecting forensic data and court-admissible evidence. We propose to explore the protocol-inherent features of VoIP in an effort to identify their forensic value, and develop a methodology for obtaining such evidence.

2.2. Forensic evaluation of personal data assistants

The development of the Handheld Evidence Recovery Operator HERO has created a new paradigm for extracting evidence from portable devices without modifying the device. We propose to develop a handheld evidence recovery operator to address the challenges created by user passwords and PINs, and a standalone tool to copy the data located on the device to an external storage medium without modifying the device.

2.3. Develop algorithms for secure covert communications using unstructured broadcast protocols and Peer-to-Peer carriers.

The third task proposes to explore unstructured broadcast protocols to implement covert channels on personal communication systems. The proposed algorithm also exploits Peer-to-Peer carriers to hide the presence of such communication channels.

1. Voice over IP Forensics

A survey of the available VoIP vendors and protocols in the United States was conducted initially. The list of vendors is located in Appendix A. The VoIP forensic study was done using phones from Vonage (http://www.vonage.com). The protocol used in this case is SIP for session maintenance and RTP for payload. Both protocols utilize UDP. The payload consists of unencrypted voice traffic. The codec used was the ITU-T g721 standard.

Session Reconstruction: We investigated the possibility of capturing and replaying a VoIP session. A laptop running ethereal, a packet capture program, in promiscuous mode was placed on the same local network as the VoIP phone. The network utilized a hub for demonstration. A switch or router capable of port monitoring can also be used. A filter to capture packets from and to the phone was specified (host 192.168.0.3). In the case where a computer based phone is used, the source and destination ports can be used to restrict the packet capture to the session.

A conversation was initiated from the VoIP phone to another VoIP phone outside the United States and the packets were captured. The same is true for incoming calls and calls placed or received from a regular circuit switched telephone. The captured packets were saved in the pcap library format for further analysis. A tool written in C parses through the capture file to extract packet data in two variations. The first run of the parser extracts all the data in the order it appears in the capture file. This was saved in a file with the extension g721 to denote the codec used. Audio Alchemy MP3 edition was used to convert the data to mp3 format. The mp3 file was played using Microsoft Windows Media Player. The playback was not clear. The second run of the parser extracted the conversation in each direction separately and placed them in separate files. This was done by using the source and destination IP address to determine the direction of flow. These files were converted to mp3. The result in this case was far better than the previous one. The source code for the Packet Parser is located in Appendix B.

An analysis on the effect of dropped packets on the reconstruction was conducted. The preliminary results show that the conversation survives up to about 30% drop in packets. The study was conducted by selecting a percentage of packets randomly and removing them from the capture file before processing. The objective of the effort was to simulate an environment where there is packet loss and to account for out of order delivery. The dropped packets were a percentage of the total number of data packets captured. The source code for the program to drop packets is located in Appendix B.

The SIP packets exchanged during a Voice over IP session contain identifiably information such as the source and destination phone numbers and ip addresses. A recovered SIP packet is located in Appendix C.

We developed a database of next-generation cell phones in commercial use around the world, and we tabulated common characteristics. This database contains information about the processor, memory, operating system, generation, protocols as well as other specifications about the phones. An Excel spreadsheet of this database is attached to this report under separate cover. This spreadsheet is titled “Phone-Specification-Database.xls”.

2. HERO Analysis Tool

1. SYSTEM REQUIREMENTS.
- Microsoft Windows XP operating system;
- Intel Pentium or compatible CPU;
- 64 MB of RAM;
- 30 MB free hard disk space;
- Microsoft Visual Studio.net 2003;
- Internet Explorer Version 6.02 or above;

2. COMPONENTS.
- User authentication required for access;
- Actions log kept for all users;
- Ability to select from multiple drives;

- Ability to sort files by known extensions, or user inputted extension;
- Search function;
    - Wildcard search ability
    - Extension search
    - Search entire drive or within specific directories
    - Keeps track of how many items were found in a specific search.
    - Program keeps all files searched for until user clears the results.
- mD5 Hash value calculation;
- View files in text/hex;
- Search for text/hex;
- Limited steganography detection;

3. USING COMPONENTS

3.1 USER AUTHENTICATION / NAME INPUT
- User enters in name, password (which will be hidden), ID number, and CASE ID number.
- User enters in name in order to keep track in the log

3.2 ACTIONS LOG
- The program keeps a detailed log of the user's actions.
- Log kept for:
    - User opens/closes the program
    - User searches for files
    - User aborts searches for files
    - User searches for text/hex in files
    - User views files in hexadecimal or plain text
    - User views the log
- Log contains:
    - User name
    - Date and time action occurred
    - What action occurred (ie. searched for text, searched for a file)
    - Outcome of the action (ie. if the text was found, file found, etc.)
- Log can be viewed by selecting "Log -> View Actions Log".

3.3 CHANGE DRIVE
- Click on the drop down tab under the "Drive Label" and select the drive you wish to access.

3.4 SELECTING FOLDERS
- Double click on the folder that you would like to view the contents of. This is located under the "Directories" label.
- To move up in a directory click on the folder that is located directly above the current folder.

3.5 FILE SORTING
- User can sort files in a given drive by selecting "File -> Type"
- User can sort by known file extensions such as images, html, etc.
- User can input a file extension to sort by.

- Done by selecting "File -> Type -> Other
- Then inputting the extension desired.

3.6 SELECTING FILES
- Double click on the file you want from the file list box. This is located under the "Files" label.

3.7 VIEWING FILES IN TEXT/HEX
- Double click on the file you want from the file list box. This brings up both the text and hex view of the file.
- To view only the text or the hex, right click on the file then:
    - "View as Plain Text" for the text view.
    - “View in Hexadecimal” for the hex view.

3.8 SEARCH FUNCTION (FILES)
- Search function will only search within specified directories.
- To change search directory select the specific folder you want to search under by double clicking the folder under the "Directories" label.
- To search entire drive select the main folder.
- The search results page will display all the files found, including paths, and how many files were found.
- To clear all search results press the "Clear Results" button, and then "OK".
- Examples of how to search.
    - Search all files on drive by searching for: .
    - Search all files of a specific extension: .extension (example .exe)
    - Search for files that contain specific keywords: keyWord

3.9 SEARCH FUNCTION (TEXT/HEX)
- The user enters the text/hex that they want to search for.
- Select the specific button either "Find Text", or "Find Hex" and the program will search for it inside that specific file.
- If the button is pressed again then the program will search for the next match. The user can continue to find further matches until there are no more.
- After all matches are found, the program will bring the user to the first match found in the program.
- The list box in the lower right hand corner will display a message with the results of your search.

3.10 STEGANOGRAPHY DETECTION
- Right click on a file in the file list box. Left click on: "Hidden Data".

• Viewing Files from the Extraction

To view the files from the extraction, the user needs to select the SD card using "File -> Open". The user can then browse through the extracted folders and files using the browser on the left side of the program. For images the files will open in the right half of the screen. HTML files will open a browser in a separate window with the file, plus it will show the text from the file in the program window. Any type of file can also be viewed in plain text or in hexadecimal. If the

user wants to view files in either of these formats they can right click on the file name and choose which one they prefer. These interpretations of the files will appear in the right half of the program screen.

- Screening Specific Types of Files

The user can also screen which types of files will be shown in the file viewer. If they want they can select "File -> Type", and then whichever type of file they want to screen. This will allow them to view a specific group of files that the user is targeting.

- Summary of Directory

The user can use this function to receive a summary of the current directory. This will enable them to determine the contents of the directory. If they choose to use this function they need to double click on the folder name, and then right click on it and choose “Get Summary”.

3.11 MD5 Hash

- Right Click on the file that you would like to compute the hash for located under the “Files” label.
- Click on “Calculate MD5 Hash”
- The resulting MD5 hash will be displayed under the MD5 Hash text box

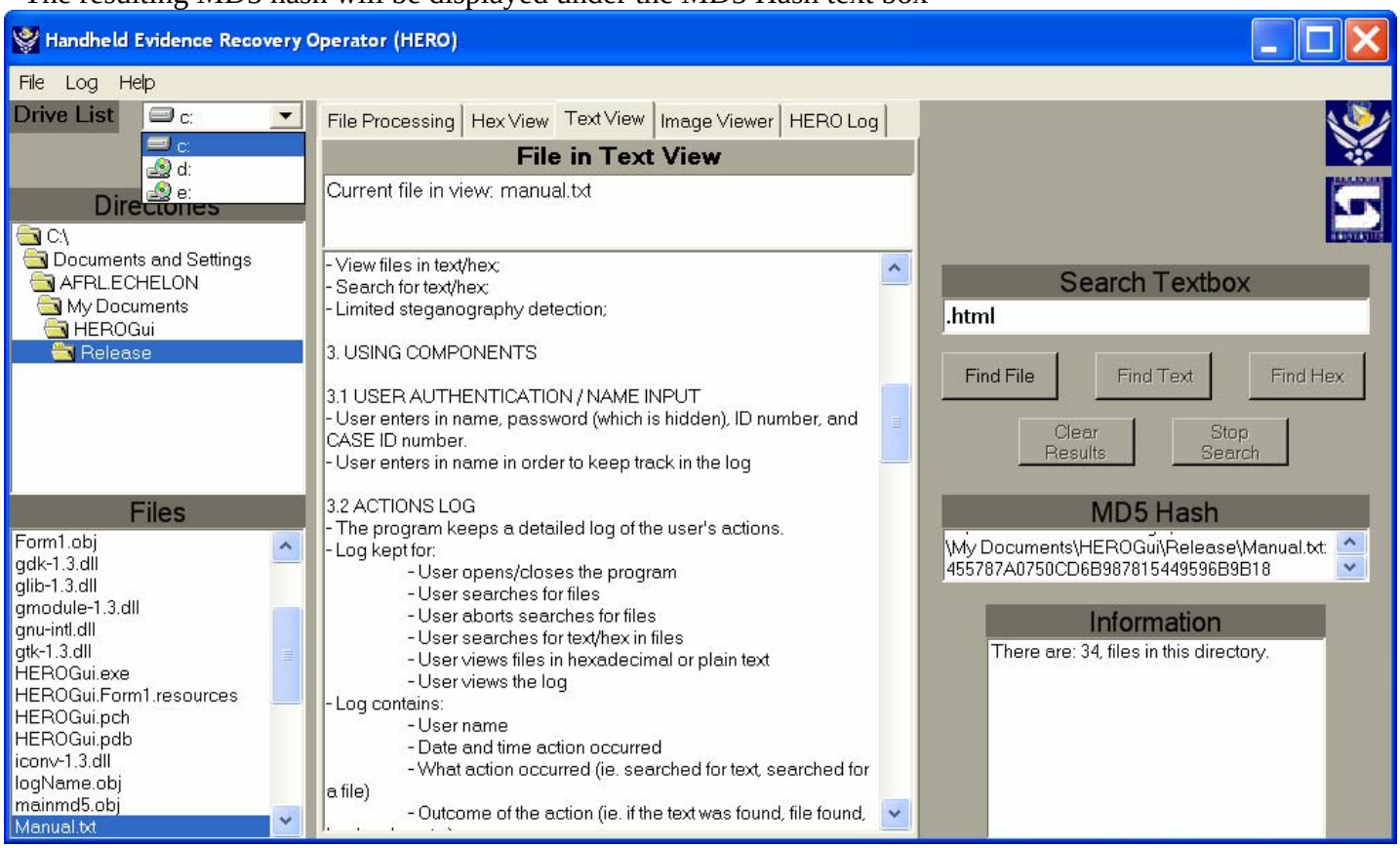

Figure 1: Screen Shot of HERO

3. Instructions for HERO Acquisition Tools

HERO has the following functionality:

- Detects the SD card and obtains its name
- Detects if the SD card has been inserted or removed
- Contains four modules
    - *Module 1 – Extracts the ROM
    - *Module 2 – Calculates the size of the object store in RAM and extracts it
    - *Module 3 – Recursively extracts files and folders
    - *Module 4 – Extracts databases
    - *Module 5 – Extracts Registry
- Can function in two modes, Standalone or Autorun
    - *Standalone – each module is self-contained and the user can execute them manually or have them run automatically using the Autorun feature in Windows CE
    - *Autorun – The autorun.exe provided with the package can execute each module in succession, if all modules are required.
- Each module displays a dialog that informs the status of that module. The dialog displays a progress bar indicating the progress. When a module finishes, it displays a message that the process is done and the process automatically closes in 10 seconds without user intervention.

Instructions for Using HERO:

General – Place the zero byte file called exists.hero in the root of the SD card. This file allows the modules to detect the SD card.

Standalone – Place the module(s) in the SD card. Insert the SD card in the target device. Browse to the SD card folder and tap on the executable module to start execution. A dialog with the progress is displayed. Wait until the process is complete and the dialog closes to remove the SD card or execute another module. If the Autorun feature is needed, rename the desired module to autorun.exe, create a folder called 2577 (for target devices with ARM processor) and place the executable in the folder. Insert the card in the device to have the module auto execute.

Autorun – Create a folder called 2577 (for target devices with ARM processor) in the SD card. Place all or some of the following files in the folder: autorun.exe, MODULE1.exe, MODULE2.exe, MODULE3.exe, MODULE4.exe and MODULE5.exe. Insert the card in the target device to have the modules execute. The modules are executed sequentially. Each module displays a dialog with its progress.

Note: Not all Windows CE based Pocket PC devices have the autorun feature enabled. The Autorun is an optional feature and may be disabled by some OEMs.

An analysis of the device before and after HERO executed confirmed that the program did not change the contents of the device. This was shown by performing an MD5 Hash on the device.

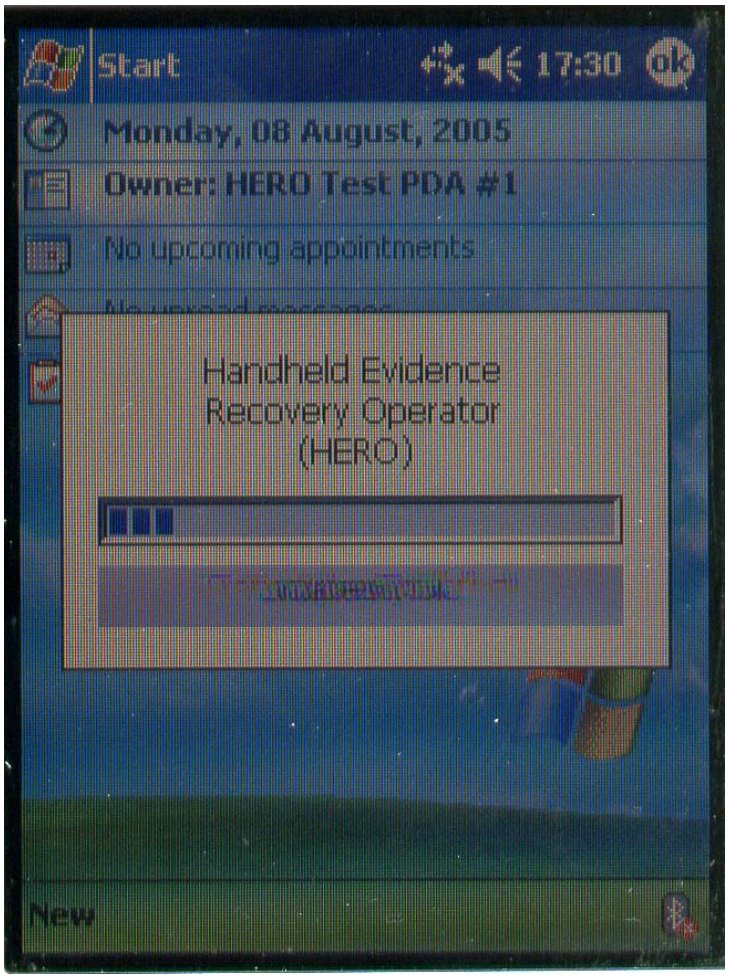

Figure 2: Screen Shot of HERO Acquisition Tool

Pocket PC 2003 Emulator:

The following are the steps to install the Pocket PC 2003 SDK, which contains the Pocket PC 2003 Emulator:

Step 1: Download and install Embedded Visual C++ 4.0 from
http://www.microsoft.com/downloads/details.aspx?familyid=1DACDB3D-50D1-41B2-A107-FA75AE960856&displaylang=en

Step 2: Download and install the Service Pack 3 for eVC++ 4.0 from
http://www.microsoft.com/downloads/details.aspx?FamilyID=5bb36f3e-5b3d-419a-9610-2fe53815ae3b&DisplayLang=en

Step 3: Download and install the Pocket PC 2003 SDK from

http://www.microsoft.com/downloads/details.aspx?FamilyID=9996b314-0364-4623-9ede-0b5fbb133652&displaylang=en

The Pocket PC 2003 Emulator can be accessed by navigating through to Start->Programs->Microsoft Pocket PC 2003 SDK->Pocket PC 2003 Emulator

Why Pocket PC 2003 Emulator?

The Pocket PC 2003 Emulator allows local drives/folders to be shared with the Emulator. This facilitates our application by providing a means to browse through the SD card.

How do we use the folder sharing feature?

After starting the emulator, we are provided with the familiar image of a Pocket PC Today screen. To share a folder, select the Folder Sharing option from the File menu on the emulator.

How do we browse the SD card in the emulator?

We use the SD Card Reader for this. Insert the SD card in the card reader.
Select the Folder sharing option in the emulator and browse to the drive letter of the SD card. Select the drive and select OK.

Navigate to the My Device on the emulator. The shared folder appears as a storage card. The contents of the SD card can now be accessed through the storage card folder on the emulator.

The source code for all of the HERO Acquisition Modules and Autorun are located in Appendix D.

Appendix E contains a dump file of all of the processes from a PocketPC platform handheld device.

Appendix F contains a paper that includes information about the HERO program and handheld device forensics.

Appendix A - Voice over IP Providers:

VoIP Providers	Web Address
Verizon VoiceWing	http://www22.verizon.com/ForYourHome/VOIP/VOIPHome.aspx
Lingo	http://www.lingo.com/voip/retail/index_retail.jsp
TeleVantage Online (Vertical Communcations)	http://www.televantageonline.com/VOIP_Info.htm
NEC Unified Solutions	http://www.necunifiedsolutions.com/cng/
Toshiba and TOTLCOM	http://www.totlcom.com/toshiba_voip.htm
Vonage	http://www.vonage.com/
ShoreTel	http://www.shoretel.com/STCorp/
iConnectHere	http://www.iconnecthere.com/nonmembers/eng/index.asp
Data Network Solutions	http://www.dnetit.com/voip.asp
BroadVoice	http://www.broadvoice.com/?AdCode=google
Net2Phone	http://www.universtelecoms.com/indexuk.htm
AT&T	http://www.att.com/voip/
Skype	http://www.skype.com
3Com	http://www.3com.com/voip/
Yak Worldcity VoIP	http://worldcity.yak.com/en/
SureVoice	http://www.surevoice.com/
ANEW Broadband	http://www.anewbroadband.com/
JustCom	http://www.justcom.biz/file.php/home
Telco Systems	http://www.telco.com/products/Access/EtoOconverters/EdgeLinkT1_Access/

VoIP Hardware Providers	Web Address
RADirect	http://www.rad-direct.com/Prod-VoIP-Hardware-Sphere.htm
VoIP Hardware Center	http://www.voip-news.com/hw/center.htm
VoIP Hardware and Equipment	http://www.callback4u.com/voice-over-ip/hardware.htm

Appendix B Source Code for Voice over IP programs
B.1 - Source Code for Packet Parser:

```
//This file has the main capture routines. It has been modified from a sample
program that demonstrates using the winpcap library

#include "main.h"
#include <windows.h>

/* prototype of the packet handler */
//void packet_handler(u_char *param, const struct pcap_pkthdr *header, const
u_char *pkt_data);

//Packet Handler for handling packets in a file

void CaptureFile_handler(u_char *param, const struct pcap_pkthdr *header,
const u_char *pkt_data)
{
      ip_header *ih;
      udp_header *uh;
      char* tdata;
      u_int ip_len;
    FILE* temp;
      u_int tlen;
      int tempVar1 = 0;
      char tempFile1[256];

      /* retireve the position of the ip header */
      ih = (ip_header *) (pkt_data +
            14); //length of ethernet header

      //get length of ip header
      ip_len = (ih->ver_ihl & 0xf) * 4;

      //retireve the position of the tcp header
      uh = (udp_header *) ((u_char*)ih + ip_len);

    //get length of udp header
      tlen = sizeof(udp_header);

      //get the position where data begins
    tdata = (char*)uh + tlen + 12;

      //calculate the total header length
      //hlen = 14+ip_len+tlen+8;

      //int i = 0;
            strcpy(tempFile1, (const char*)param);

            temp = fopen((const char*)tempFile1, "a+");

            fprintf(temp, "%s", tdata);

            fclose(temp);

      return;
```

```
}


int main(int argc, char* argv[])
{
      pcap_t *fp = NULL;
      char tempFile1[256];
    char errbuf[PCAP_ERRBUF_SIZE];

      GetCurrentDirectory(256, tempFile1);

      if ( (fp = pcap_open_offline(argv[1], errbuf) ) == NULL)
      {
            return 0;
      }

      pcap_loop(fp, 0, CaptureFile_handler, (u_char*)argv[2]);

      return 1;
}
```

B.2 - Source Code for Program to Drop Packets:

```
//This file has the main capture routines. It has been modified from a sample
program that demonstrates using the winpcap library
#include <vector>
#include <algorithm>
#include "rand.h"
#include "main.h"

//#include <windows.h>

/* prototype of the packet handler */
//void packet_handler(u_char *param, const struct pcap_pkthdr *header, const
u_char *pkt_data);

//Packet Handler for handling packets in a file

int NumPackets = 0;
int TotPackets = 0;
int PacketCount = 0;
std::vector<int> DropPackets;
std::vector<int>::iterator temp;

void CaptureFile_handler(u_char *param, const struct pcap_pkthdr *header,
const u_char *pkt_data)
{
      PacketCount++;
      if(temp != DropPackets.end())
      {
            //printf("%d\n", PacketCount);
            if(*temp == PacketCount)
            {
                  //printf("%d\n", PacketCount);
```

```
                temp++;
                return;
            }
        }


        ip_header *ih;
        udp_header *uh;
        char* tdata;
        u_int ip_len;
      FILE* temp;
        u_int tlen;
        int tempVar1 = 0;
        char tempFile1[256];

        /* retireve the position of the ip header */
        ih = (ip_header *) (pkt_data +
              14); //length of ethernet header

        //get length of ip header
        ip_len = (ih->ver_ihl & 0xf) * 4;

        //retireve the position of the tcp header
        uh = (udp_header *) ((u_char*)ih + ip_len);

      //get length of udp header
        tlen = sizeof(udp_header);

        //get the position where data begins
      tdata = (char*)uh + tlen + 12;

        //calculate the total header length
        //hlen = 14+ip_len+tlen+8;

        //int i = 0;
              strcpy(tempFile1, (const char*)param);

              temp = fopen((const char*)tempFile1, "a+");

              fprintf(temp, "%s", tdata);

              fclose(temp);

        return;
}

void CountPackets_handler(u_char *param, const struct pcap_pkthdr *header,
const u_char *pkt_data)
{
        TotPackets++;
        return;
}

void q_sort(int left, int right)
{
  int pivot;
  int l_hold;
```

```
  int r_hold;

  l_hold = left;
  r_hold = right;
  pivot = DropPackets.at(left);
  while (left < right)
  {
    while ((DropPackets.at(right) >= pivot) && (left < right))
      right--;
    if (left != right)
    {
      DropPackets.at(left) = DropPackets.at(right);
      left++;
    }
    while ((DropPackets.at(left) <= pivot) && (left < right))
      left++;
    if (left != right)
    {
      DropPackets.at(right) = DropPackets.at(left);
      right--;
    }
  }
  DropPackets.at(left) = pivot;
  pivot = left;
  left = l_hold;
  right = r_hold;
  if (left < pivot)
    q_sort(left, pivot-1);
  if (right > pivot)
    q_sort(pivot+1, right);
}

void quickSort(int array_size)
{
  q_sort(0, array_size-1);
}

int main(int argc, char* argv[])
{
      long int RandomVal = 0;
      int TheCount = 0;
      int TheCount1 = 0;
      pcap_t *fp = NULL;
      char tempFile1[256];
    char errbuf[PCAP_ERRBUF_SIZE];

      GetCurrentDirectory(256, tempFile1);

      if ( (fp = pcap_open_offline(argv[1], errbuf) ) == NULL)
      {
            printf("Could Not Open File %s\n", argv[1]);
            return 0;
      }

      pcap_loop(fp, 0, CountPackets_handler, (u_char*)argv[0]);

      NumPackets = atoi(argv[3]);
```

```
//printf("\nTotal Packets: %d\nPackets to Drop: %d\n", TotPackets,
NumPackets);
      if(NumPackets == 101)
      {
      }

      else
      {

      NumPackets = (NumPackets*TotPackets)/100;

      TheCount = NumPackets;
      //TheCount1 = NumPackets;

      printf("%d\n%d\n", NumPackets, TheCount);

      randctx tempRand;

      randinit(&tempRand, FALSE);

      while(NumPackets)
      {

            RandomVal = rand(&tempRand);
            if(RandomVal >= 1)
            {
            while(RandomVal >= TotPackets)
                  {
                RandomVal = RandomVal - (TotPackets-1);
                  }
            }
            else
            {
                  while(RandomVal < 1)
                  {
                        RandomVal = RandomVal + (TotPackets-1);
                  }
            }
            //printf("%d\n", RandomVal);

            std::vector<int>::iterator temp1;

            temp1 = find(DropPackets.begin(), DropPackets.end(), RandomVal);
        if(temp1 == DropPackets.end())
            {
            DropPackets.push_back(RandomVal);
            NumPackets--;
                  TheCount1++;
            }
      }

      //quickSort(TheCount1);
      sort(DropPackets.begin(), DropPackets.end());

      printf("Total Packets: %d\nPackets to Drop: %d", TotPackets,
TheCount1);
      }
```

```
	//for(temp = DropPackets.begin();temp != DropPackets.end(); temp++)
		//printf("%d\n", *temp);
	temp = DropPackets.begin();

	if ( (fp = pcap_open_offline(argv[1], errbuf) ) == NULL)
	{
		printf("Could Not Open File %s\n", argv[1]);
		return 0;
	}

	printf("%d\n%d\n", NumPackets, TheCount);


	pcap_loop(fp, 0, CaptureFile_handler, (u_char*)argv[2]);



	return 1;
}
```

Appendix C – SIP Packet

The SIP packets exchanged by the end points reveal the source and destination phone number, ip address and caller id. The gateway used for packet forwarding is also available in the packet.

The following SIP packet payload shows the caller ID, phone number and IP address of the destination in bold.

ACK sip:13157039646@216.115.25.19:5061 SIP/2.0
From: "315-299-2459"<sip:13152992459@atlas5.atlas.vonage.net:5061;user=phone>;tag=c0a80003-13c5-42453181-238ec570-7fbd
To: "**rukmanis**"<sip:**13157039646**@atlas5.atlas.vonage.net:5061;user=phone>;tag=594026082
Call-ID: 9451d2ec-20106-1111830913-35793553-12860341970211850000000-0@192.168.0.3
CSeq: 2 ACK
Via: SIP/2.0/UDP 192.168.0.3:5061;branch=z9hG4b K-42453184-238ecf98-5ab9
Max-Forwards: 70
Contact: <sip:13152992459@192.168.0.3:5061;transport=UDP;user=phone>
Route: <sip:13157039646@216.115.27.29:5060>
Route: <sip:13157039646@216.115.25.198:5061>
Route: <sip:13157039646@**196.12.47.41:43780**>
Proxy-Authorization: Digest username="13152992459", realm="216.115.25.19", nonce="1685999676", uri="sip:13157039646@atlas5.atlas.vonage.net:5061;user=phone", response="15860f8a91a2c19c6589d2ccf7df7bcb", algorithm=MD5
Content-Length: 0

Appendix D - Source Code for HERO Acquisition Tool
D.1 – Source code for Autorun

```
#include <windows.h>

BOOL DirNavigate(LPTSTR searchPath)
{


      HANDLE fileHandle;
    WIN32_FIND_DATA findData;


      //TCHAR searchPath[MAX_PATH];
      TCHAR ThePath[MAX_PATH];

      lstrcpy (searchPath, TEXT("\\*.*"));


      fileHandle = FindFirstFile (searchPath, &findData);

      if (fileHandle != INVALID_HANDLE_VALUE)
      {
            do
            {

                  lstrcpy(searchPath, TEXT("\\"));

                  lstrcat(searchPath, findData.cFileName);

                  //MessageBox(NULL, searchPath, TEXT("Search"), MB_OK);

                  if (findData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)
                  {
                        //if (findData.dwFileAttributes &
FILE_ATTRIBUTE_TEMPORARY)
                        //{

                              lstrcpy(ThePath, TEXT("\\"));
                              lstrcat(ThePath, findData.cFileName);
                              lstrcat(ThePath, TEXT("\\exists.hero"));
                              //MessageBox(NULL, ThePath, TEXT("Search"),
MB_OK);

                              HANDLE temp = CreateFile(ThePath, GENERIC_READ,
0, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
                              if (temp != INVALID_HANDLE_VALUE)
                              {
                                    CloseHandle(temp);
                                    FindClose (fileHandle);
                                    //MessageBox(NULL, searchPath,
TEXT("SD"), MB_OK);
                                    return TRUE;
                              }
                        //}
```

```
                  }

            }
            while (FindNextFile (fileHandle, &findData));

      }
      FindClose (fileHandle);

      return FALSE;

}

int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
                    LPWSTR lpCmdLine, int nCmdShow)
{

      if(lstrcmp(lpCmdLine, TEXT("uninstall")) == 0) return 0;

      if((lstrcmp(lpCmdLine, TEXT("install")) == 0) || (lstrcmp(lpCmdLine,
TEXT("")) == 0))
      {
            TCHAR SDPath[MAX_PATH];
            if(DirNavigate(SDPath) == FALSE)
                  return 0;
            //MessageBox(NULL, SDPath, TEXT("Hi"), MB_OK);


PROCESS_INFORMATION pi;
int rc = 0;

TCHAR ExePath[MAX_PATH];

lstrcpy(ExePath, SDPath);
lstrcat(ExePath, _T("\\2577\\MODULE1.exe"));

rc = CreateProcess (ExePath, SDPath, NULL, NULL, FALSE,
                    0, NULL, NULL, NULL, &pi);
WaitForSingleObject(pi.hProcess, INFINITE);
if (rc) {
    CloseHandle (pi.hThread);
    CloseHandle (pi.hProcess);
}

rc = 0;

lstrcpy(ExePath, SDPath);
lstrcat(ExePath, _T("\\2577\\MODULE2.exe"));


rc = CreateProcess (ExePath, SDPath, NULL, NULL, FALSE,
                    0, NULL, NULL, NULL, &pi);
WaitForSingleObject(pi.hProcess, INFINITE);
if (rc) {
    CloseHandle (pi.hThread);
    CloseHandle (pi.hProcess);
}
```

```
rc = 0;

lstrcpy(ExePath, SDPath);
lstrcat(ExePath, _T("\\2577\\MODULE3.exe"));


rc = CreateProcess (ExePath, SDPath, NULL, NULL, FALSE,
                    0, NULL, NULL, NULL, &pi);
WaitForSingleObject(pi.hProcess, INFINITE);
if (rc) {
    CloseHandle (pi.hThread);
    CloseHandle (pi.hProcess);
}

rc = 0;

lstrcpy(ExePath, SDPath);
lstrcat(ExePath, _T("\\2577\\MODULE4.exe"));

rc = CreateProcess (ExePath, SDPath, NULL, NULL, FALSE,
                    0, NULL, NULL, NULL, &pi);
WaitForSingleObject(pi.hProcess, INFINITE);
if (rc) {
    CloseHandle (pi.hThread);
    CloseHandle (pi.hProcess);
}

rc = 0;


/*rc = CreateProcess (TEXT ("\\Storage Card\\MODULE5.exe"), SDPath, NULL,
NULL, FALSE,
                    0, NULL, NULL, NULL, &pi);
WaitForSingleObject(pi.hProcess, INFINITE);
if (rc) {
    CloseHandle (pi.hThread);
    CloseHandle (pi.hProcess);
}*/
      }
return 1;

}
```

D.2 – Source code for Module 1

```
#include "stdafx.h"
#include "main.h"
#include "Dlg.h"
#include <windows.h>
#include "pkfuncs.h"


////////////////////////////////////////////////////////////////////////
// CFileApp

BEGIN_MESSAGE_MAP(CFileApp, CWinApp)
```

```
END_MESSAGE_MAP()

//////////////////////////////////////////////////////////////////////////////
// CFileApp construction

CFileApp::CFileApp()
	: CWinApp()
{
	pDlg = NULL;
}

CFileApp::~CFileApp()
{
	delete pDlg;
}


CFileApp theApp;



BOOL CFileApp::InitInstance()
{
	TCHAR temp[MAX_PATH];
	//HINSTANCE SDdll;
	//LPFNDLLFUNC1 SDadrs;

	if ((lstrcmp(m_lpCmdLine, TEXT("")) == 0) || (lstrcmp(m_lpCmdLine,
TEXT("install")) == 0) || (lstrcmp(m_lpCmdLine, TEXT("uninstall")) == 0))
	{
		//lstrcpy(temp, _T("\\Storage Card"));
		return FALSE;

	}



	//else
	//{
		lstrcpy(temp, m_lpCmdLine);
	//}

	int myProgressPos;

	pDlg = new CFileDlg();

	pDlg->m_bFullScreen = FALSE;
	int iResponse = pDlg->Create(IDD_FILE_DIALOG, NULL);

	//Sleep(2000);

	pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting ROM..."));

	//lstrcpy(temp, m_lpCmdLine);

	if(!ROMReader(pDlg, temp))
```

```
	{
		Sleep(2000);
		delete this;
		return FALSE;
	}

	myProgressPos = pDlg->myPctrl.GetPos();
	if (myProgressPos < 10)
	{
		for(int tInker = myProgressPos; tInker < 10; tInker++)
		{
			pDlg->myPctrl.StepIt();
			Sleep(5);
		}
	}

	pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting ROM...Done"));

	Sleep(2000);


	delete this;


	// Since the dialog has been closed, return FALSE so that we exit the
	//  application, rather than start the application's message pump.
	return FALSE;
}


int CFileApp::ROMReader(CFileDlg* pDlg, TCHAR SDPath[MAX_PATH])
{
	lstrcat(SDPath, TEXT("\\Rom"));

	BOOL temp = CreateDirectory(SDPath, NULL);

	lstrcat(SDPath, _T("\\Rom.bin"));

	if(!temp)
	{
		FILE* existFile;
		existFile = _wfopen(SDPath, _T("r"));
		if (existFile != NULL)
		{
			fclose(existFile);
			pDlg->SetDlgItemText(IDC_EDIT1, _T("ROM image exists"));
			return 0;
		}
	}

	//lstrcat(SDPath, TEXT("\\RomImage.bin"));

	DWORD objStoreLoc = 0x80000000;

	SetProcPermissions(0xFFFF);
	#define SIZE (1024 * 1024 * 4)
```

```
	BYTE* lpv;
	BOOL bRet = TRUE;
	int Offset = 0;

	FILE* output;

	while ((objStoreLoc+Offset) < 0x82000000)
	{

		lpv = (BYTE*) VirtualAlloc(0, SIZE, MEM_RESERVE, PAGE_NOACCESS);

		bRet = VirtualCopy(lpv, (void*)(objStoreLoc+Offset), SIZE,
PAGE_READWRITE | PAGE_NOCACHE);

		if (bRet)
		{
			output = _wfopen(SDPath, _T("a+b"));
			fwrite(lpv, sizeof(BYTE), SIZE, output);
			fclose(output);
		}

		VirtualFree(lpv, 0, MEM_RELEASE);

		Offset = Offset + SIZE;
		pDlg->myPctrl.StepIt();
	}

	return 1;
}
```

D.3 – Source code for Module 2

```
#include "stdafx.h"
#include "main.h"
#include "Dlg.h"
#include <windows.h>
#include "pkfuncs.h"


////////////////////////////////////////////////////////////////////////////
// CFileApp

BEGIN_MESSAGE_MAP(CFileApp, CWinApp)

END_MESSAGE_MAP()

////////////////////////////////////////////////////////////////////////////
// CFileApp construction

CFileApp::CFileApp()
	: CWinApp()
{
	pDlg = NULL;
}
```

```
CFileApp::~CFileApp()
{
	delete pDlg;
}


CFileApp theApp;



BOOL CFileApp::InitInstance()
{

	//if(lstrcmp(m_lpCmdLine, TEXT("uninstall")) == 0) return 0;

	//if(lstrcmp(m_lpCmdLine, TEXT("install")) == 0)
	//{


	TCHAR temp[MAX_PATH];

	if ((lstrcmp(m_lpCmdLine, TEXT("")) == 0) || (lstrcmp(m_lpCmdLine,
TEXT("install")) == 0) || (lstrcmp(m_lpCmdLine, TEXT("uninstall")) == 0))
	{
		//lstrcpy(temp, _T("\\Storage Card"));
		return FALSE;

	}



	//else
	//{
		lstrcpy(temp, m_lpCmdLine);
	//}

	int myProgressPos;

	pDlg = new CFileDlg();

	pDlg->m_bFullScreen = FALSE;
	int iResponse = pDlg->Create(IDD_FILE_DIALOG, NULL);

	pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Object Store..."));

	if(!ObjReader(pDlg, temp))
	{
		Sleep(2000);
		delete this;
		return FALSE;
	}

	myProgressPos = pDlg->myPctrl.GetPos();
	if (myProgressPos < 10)
	{
		for(int tInker = myProgressPos; tInker < 10; tInker++)
```

```
			{
				pDlg->myPctrl.StepIt();
				Sleep(5);
			}
		}

		pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Object Store...Done"));

		Sleep(2000);


		delete this;

		//}


		// Since the dialog has been closed, return FALSE so that we exit the
		//  application, rather than start the application's message pump.
		return FALSE;
}


int CFileApp::ObjReader(CFileDlg* pDlg, TCHAR SDPath[MAX_PATH])
{

		lstrcat(SDPath, _T("\\ObjStore"));

		BOOL SDDir = CreateDirectory(SDPath, NULL);

		lstrcat(SDPath, _T("\\ObjStore.bin"));

		if(!SDDir)
		{
			FILE* existFile;
			existFile = _wfopen(SDPath, _T("r"));
			if (existFile != NULL)
			{
				fclose(existFile);
				pDlg->SetDlgItemText(IDC_EDIT1, _T("Object Store image
exists"));
				return 0;
			}
		}

		DWORD objStoreLoc = GetFSHeapInfo();


		SetProcPermissions(0xFFFF);
		#define SIZE (1024*1024*4)

		BYTE* lpv;
		BOOL bRet = TRUE;
		int Offset = 0;

		STORE_INFORMATION StoreInfoVar;
		FILE* output;
		GetStoreInformation(&StoreInfoVar);
```

```
	DWORD temp = objStoreLoc + StoreInfoVar.dwStoreSize;


	while ((objStoreLoc+Offset) < temp)
	{
		lpv = (BYTE*) VirtualAlloc(0, SIZE, MEM_RESERVE, PAGE_NOACCESS);

		if(!VirtualCopy(lpv, (void*)(objStoreLoc+Offset), SIZE,
PAGE_READWRITE | PAGE_NOCACHE))
		{
			VirtualFree(lpv, 0, MEM_RELEASE);
			return 1;
		}

		output = _wfopen(SDPath, _T("a+b"));
		fwrite(lpv, sizeof(BYTE), SIZE, output);
		fclose(output);

		VirtualFree(lpv, 0, MEM_RELEASE);

		Offset = Offset + SIZE;

		pDlg->myPctrl.StepIt();

	}

	return 1;
}
```

D.4 – Source code for Module 3

```
#include "stdafx.h"
#include "main.h"
#include "Dlg.h"
#include <windows.h>
//#include "md5.h"


/////////////////////////////////////////////////////////////////////////////
// CFileApp

BEGIN_MESSAGE_MAP(CFileApp, CWinApp)

END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CFileApp construction

CFileApp::CFileApp()
	: CWinApp()
{
	pDlg = NULL;
}

CFileApp::~CFileApp()
```

```
{
	delete pDlg;
}


CFileApp theApp;



BOOL CFileApp::InitInstance()
{


	if ((lstrcmp(m_lpCmdLine, TEXT("")) == 0) || (lstrcmp(m_lpCmdLine,
TEXT("install")) == 0) || (lstrcmp(m_lpCmdLine, TEXT("uninstall")) == 0))
	{
		//lstrcpy(temp, _T("\\Storage Card"));
		return FALSE;

	}

		TCHAR temp[MAX_PATH];


	//else
	//{
		lstrcpy(temp, m_lpCmdLine);
	//}


	int myProgressPos;

	pDlg = new CFileDlg();

	pDlg->m_bFullScreen = FALSE;
	int iResponse = pDlg->Create(IDD_FILE_DIALOG, NULL);



	//lstrcpy(temp, _T("\\Storage Card"));
	lstrcat(temp, _T("\\Files"));

	BOOL SDDir = CreateDirectory(temp, NULL);

	if(!SDDir)
	{
		pDlg->SetDlgItemText(IDC_EDIT1, _T("Files folder exists"));
		Sleep(2000);
		delete this;
		return FALSE;
	}

	SYSTEMTIME TheTime;
	GetSystemTime(&TheTime);
	FILE* timeFile;
	timeFile = fopen("\\Storage Card\\SystemTime.txt", "a+");
	fprintf(timeFile, "%04d-%02d-%02d %02d:%02d:%02d.%03d\t\t",
```

```
                TheTime.wYear, TheTime.wMonth, TheTime.wDay,
                TheTime.wHour, TheTime.wMinute,
TheTime.wSecond,TheTime.wMilliseconds);
      fclose(timeFile);

      Sleep(2000);

      pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Files..."));

      TheFunction(pDlg, temp);

      myProgressPos = pDlg->myPctrl.GetPos();
      if (myProgressPos < 400)
      {
            for(int tInker = myProgressPos; tInker < 400; tInker++)
            {
                  pDlg->myPctrl.StepIt();
                  Sleep(5);
            }
      }

      pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Files...Done"));

      Sleep(2000);


      delete this;


      // Since the dialog has been closed, return FALSE so that we exit the
      //  application, rather than start the application's message pump.
      return FALSE;
}


int CFileApp::TheFunction(CFileDlg* pDlg, TCHAR szName[MAX_PATH])
{

      //TCHAR szName[MAX_PATH];

      int j = 0;

      //lstrcpy(szName, TEXT("\\Storage Card\\Files"));

      FILE *Logfile;

      Logfile = fopen("\\Storage Card\\Log.txt", "a+");

      fprintf(Logfile, "File Name\t\tDirectory\t\tCreation Time\t\tLast
Access Time\t\tLast Write Time\n\n");

      if (DirNavigate(TEXT("\\"), szName, 0, pDlg, Logfile))
      {

            /*lstrcat(szName, TEXT("\\Windows"));
            CreateDirectory(szName, NULL);
            if (DirNavigate(TEXT("\\Windows\\"), szName, 1, pDlg, Logfile))
```

```
            {

            }*/
      }
      fclose(Logfile);

      return 1;

}

bool CFileApp::DirNavigate(TCHAR dirName[MAX_PATH], TCHAR
storedirName[MAX_PATH], int i, CFileDlg* pDlg, HANDLE Logfile)
{


      HANDLE fileHandle;
    WIN32_FIND_DATA findData;


      TCHAR searchPath[MAX_PATH];

      lstrcpy (searchPath, dirName);
      lstrcat (searchPath, TEXT("*.*"));

      // TEST CODE

      //MessageBox(NULL, searchPath, TEXT("hello"), MB_OK);

      //lstrcpy (searchPath, storedirName);

      //MessageBox(NULL, dirName, TEXT("Hello"), MB_OK);

      //END TEST CODE

      fileHandle = FindFirstFile (searchPath, &findData);

      if (fileHandle != INVALID_HANDLE_VALUE)
      {
            do
            {

                  lstrcpy(searchPath, dirName);

                  lstrcat(searchPath, findData.cFileName);

                  //TEST CODE

                  //MessageBox(NULL, searchPath, TEXT("Source"), MB_OK);

                  //END TEST CODE

                  TCHAR destPath[MAX_PATH];

                  lstrcpy(destPath, storedirName);
                  lstrcat(destPath, TEXT("\\"));
                  lstrcat(destPath, findData.cFileName);
```

```
                  //TEST CODE

                  //MessageBox(NULL, destPath, TEXT("Dest"), MB_OK);

                  //END TEST CODE

                  if (findData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)
                  {
                        if (lstrcmp(searchPath, TEXT("\\Storage Card")) != 0)
                        {
                              //if (lstrcmp(searchPath, TEXT("\\Windows")) !=
0)
                              //{
                                    CreateDirectory(destPath, NULL);
                                    lstrcat(searchPath, TEXT("\\"));
                                    DirNavigate(searchPath, destPath, i,
pDlg, Logfile);
                              //}
                              /*if (i == 1)
                              {
                                    CreateDirectory(destPath, NULL);
                                    lstrcat(searchPath, TEXT("\\"));
                                    DirNavigate(searchPath, destPath, i,
pDlg, Logfile);
                              }*/
                        }

                  }
                  if (!(findData.dwFileAttributes &
FILE_ATTRIBUTE_DIRECTORY))
                  {
                        //pDlg->SetDlgItemText(IDC_EDIT1,findData.cFileName);

                        //CopyFile(searchPath, destPath, 0);

                        //if (i == 1)
                        //{
                              pDlg-
>SetDlgItemText(IDC_EDIT1,findData.cFileName);
                              CopyFile(searchPath, destPath, 0);
                        //}

                        //if (lstrcmp(searchPath, TEXT("\\Windows")) != 0)
                        //{
                              //pDlg-
>SetDlgItemText(IDC_EDIT1,findData.cFileName);
                              //CopyFile(searchPath, destPath, 0);
                        //}

                        fprintf(Logfile, "%ls\t%ls\t", findData.cFileName,
searchPath);
                        SYSTEMTIME systime;
                        FileTimeToSystemTime(&findData.ftCreationTime,
&systime);
                        fprintf(Logfile, "%04d-%02d-%02d
%02d:%02d:%02d.%03d\t",
                              systime.wYear, systime.wMonth, systime.wDay,
```

```
                              systime.wHour, systime.wMinute,
systime.wSecond,systime.wMilliseconds);
                        FileTimeToSystemTime(&findData.ftLastAccessTime,
&systime);
                        fprintf(Logfile, "%04d-%02d-%02d
%02d:%02d:%02d.%03d\t",
                              systime.wYear, systime.wMonth, systime.wDay,
                              systime.wHour, systime.wMinute,
systime.wSecond,systime.wMilliseconds);
                        FileTimeToSystemTime(&findData.ftLastWriteTime,
&systime);
                        fprintf(Logfile, "%04d-%02d-%02d
%02d:%02d:%02d.%03d\n",
                              systime.wYear, systime.wMonth, systime.wDay,
                              systime.wHour, systime.wMinute,
systime.wSecond,systime.wMilliseconds);

                  }
            }
            while (FindNextFile (fileHandle, &findData));

      }
      FindClose (fileHandle);
      pDlg->myPctrl.StepIt();

      return 1;

}
```

D.5 – Source code for Module 4

```
#include "stdafx.h"
#include "main.h"
#include "Dlg.h"
#include <windows.h>


/////////////////////////////////////////////////////////////////////////////
// CFileApp

BEGIN_MESSAGE_MAP(CFileApp, CWinApp)

END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CFileApp construction

CFileApp::CFileApp()
      : CWinApp()
{
      pDlg = NULL;
}

CFileApp::~CFileApp()
```

```
{
	delete pDlg;
}


CFileApp theApp;



BOOL CFileApp::InitInstance()
{


	TCHAR temp[MAX_PATH];

		if ((lstrcmp(m_lpCmdLine, TEXT("")) == 0) ||
(lstrcmp(m_lpCmdLine, TEXT("install")) == 0) || (lstrcmp(m_lpCmdLine,
TEXT("uninstall")) == 0))
	{
		//lstrcpy(temp, _T("\\Storage Card"));
		return FALSE;

	}

	//else
	//{
		lstrcpy(temp, m_lpCmdLine);
	//}

	int myProgressPos;

	pDlg = new CFileDlg();

	pDlg->m_bFullScreen = FALSE;

	int iResponse = pDlg->Create(IDD_FILE_DIALOG, NULL);

	//pDlg->SetDlgItemText(IDC_EDIT1, _T("To the HERO-Mobile..."));

	lstrcat(temp, _T("\\Database"));

    BOOL SDDir = CreateDirectory(temp, NULL);

	pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Databases..."));

	Sleep(1000);

	if (!SDDir)
	{
		pDlg->SetDlgItemText(IDC_EDIT1, _T("Database folder exists"));
		Sleep(2000);
		delete this;
		return FALSE;
	}

	lstrcat(temp, _T("\\Databases.cdb"));
```

```
	DBFunction(pDlg, temp);

	myProgressPos = pDlg->myPctrl.GetPos();
	if (myProgressPos < 100)
	{
		for(int tInker = myProgressPos; tInker < 100; tInker++)
		{
			pDlg->myPctrl.StepIt();
			Sleep(5);
		}
	}

	pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Databases...Done"));

	Sleep(2000);


	delete this;


	//}


	// Since the dialog has been closed, return FALSE so that we exit the
	//  application, rather than start the application's message pump.
	return FALSE;
}

void DumpDatabaseByName(HANDLE h, HANDLE destDBhandle)
{


    CEOID dbid=0;

      CEPROPVAL *props=NULL;
    DWORD bufsize=0;

    CEOID rid;
    WORD nProps;

    while (0!=(rid= CeReadRecordProps(h, CEDB_ALLOWREALLOC, &nProps, NULL,
(BYTE**)&props, &bufsize)))
    {

            for (int i=0 ; i<nProps ; i++)
            {
            CeWriteRecordProps (destDBhandle, 0, 1, &props[i]);
            }
    }
    LocalFree(props);

}

int CFileApp::DBFunction(CFileDlg *pDlg, TCHAR SDPath[MAX_PATH])
{
	HANDLE hDBList;
	CEOID oidDB;
```

```
	CEOID destoidDB;
	CEOIDINFO poidInfo;

	CEGUID guidVol;
	CEGUID destGUID;
	HANDLE DBhandle;
	HANDLE destDBhandle;

	char temp = '\n';

	CREATE_SYSTEMGUID(&guidVol);
	CeMountDBVol (&destGUID, SDPath, OPEN_ALWAYS);

	hDBList = CeFindFirstDatabaseEx(&guidVol, 0);


	if (hDBList != INVALID_HANDLE_VALUE)
	{

		oidDB = CeFindNextDatabaseEx(hDBList, &guidVol);

		while(oidDB)
		{

			DBhandle = CeOpenDatabaseEx(&guidVol, &oidDB, NULL, 0,
CEDB_AUTOINCREMENT, NULL);


			if (DBhandle != INVALID_HANDLE_VALUE)
			{
				CeOidGetInfoEx(&guidVol, oidDB, &poidInfo);

				pDlg->SetDlgItemText(IDC_EDIT1,(LPCTSTR)
poidInfo.infDatabase.szDbaseName);

				destoidDB = CeCreateDatabaseEx(&destGUID,
&poidInfo.infDatabase);

				destDBhandle = CeOpenDatabaseEx (&destGUID,
&destoidDB, NULL, 0, 0, NULL);

				DumpDatabaseByName(DBhandle, destDBhandle);
				pDlg->myPctrl.StepIt();

				CloseHandle(destDBhandle);

				CloseHandle(DBhandle);

			}
			oidDB = CeFindNextDatabaseEx(hDBList, &guidVol);
		}

		CloseHandle(hDBList);

	}

	CeUnmountDBVol(&destGUID);
```

```
	return 1;
}
```

D.6 – Source code for Module 5

```
#include "stdafx.h"
#include "main.h"
#include "Dlg.h"
#include "pwinreg.h"
#include <windows.h>


//////////////////////////////////////////////////////////////////////////
// CFileApp

BEGIN_MESSAGE_MAP(CFileApp, CWinApp)

END_MESSAGE_MAP()

//////////////////////////////////////////////////////////////////////////
// CFileApp construction

CFileApp::CFileApp()
	: CWinApp()
{
	pDlg = NULL;
}

CFileApp::~CFileApp()
{
	delete pDlg;
}


CFileApp theApp;



BOOL CFileApp::InitInstance()
{


TCHAR temp[MAX_PATH];

		if (lstrcmp(m_lpCmdLine, TEXT("uninstall")) == 0)
	{
		//lstrcpy(temp, _T("\\Storage Card"));
		return FALSE;

	}

	if(DirNavigate(temp) == FALSE)
			return 0;

	int myProgressPos;
```

```
  pDlg = new CFileDlg();

  pDlg->m_bFullScreen = FALSE;

  int iResponse = pDlg->Create(IDD_FILE_DIALOG, NULL);

  //pDlg->SetDlgItemText(IDC_EDIT1, _T("To the HERO-Mobile..."));

  lstrcat(temp, _T("\\Registry"));

BOOL SDDir = CreateDirectory(temp, NULL);

  pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Registry..."));

  Sleep(1000);

  if (!SDDir)
  {
        pDlg->SetDlgItemText(IDC_EDIT1, _T("Registry folder exists"));
        Sleep(2000);
        delete this;
        return FALSE;
  }

  lstrcat(temp, _T("\\Registry.reg"));

  RegCopyFile(temp);

  //RegFunction(pDlg);

  //RegSaveKey(HKEY_CLASSES_ROOT, "\\Storage Card\\temp.reg", NULL);

  myProgressPos = pDlg->myPctrl.GetPos();
  if (myProgressPos < 100)
  {
        for(int tInker = myProgressPos; tInker < 100; tInker++)
        {
              pDlg->myPctrl.StepIt();
              Sleep(5);
        }
  }

  pDlg->SetDlgItemText(IDC_EDIT1, _T("Extracting Registry...Done"));

  Sleep(2000);


  delete this;


  //}


  // Since the dialog has been closed, return FALSE so that we exit the
  //  application, rather than start the application's message pump.
  return FALSE;
```

```
}

BOOL CFileApp::DirNavigate(LPTSTR searchPath)
{


      HANDLE fileHandle;
    WIN32_FIND_DATA findData;


      //TCHAR searchPath[MAX_PATH];
      TCHAR ThePath[MAX_PATH];

      lstrcpy (searchPath, TEXT("\\*.*"));


      fileHandle = FindFirstFile (searchPath, &findData);

      if (fileHandle != INVALID_HANDLE_VALUE)
      {
            do
            {

                  lstrcpy(searchPath, TEXT("\\"));

                  lstrcat(searchPath, findData.cFileName);

                  //MessageBox(NULL, searchPath, TEXT("Search"), MB_OK);

                  if (findData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)
                  {
                        //if (findData.dwFileAttributes &
FILE_ATTRIBUTE_TEMPORARY)
                        //{

                              lstrcpy(ThePath, TEXT("\\"));
                              lstrcat(ThePath, findData.cFileName);
                              lstrcat(ThePath, TEXT("\\exists.hero"));
                              //MessageBox(NULL, ThePath, TEXT("Search"),
MB_OK);

                              HANDLE temp = CreateFile(ThePath, GENERIC_READ,
0, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
                              if (temp != INVALID_HANDLE_VALUE)
                              {
                                    CloseHandle(temp);
                                    FindClose (fileHandle);
                                    //MessageBox(NULL, searchPath,
TEXT("SD"), MB_OK);
                                    return TRUE;
                              }
                        //}

                  }

            }
            while (FindNextFile (fileHandle, &findData));
```

```
        }
        FindClose (fileHandle);

        return FALSE;

}
```

Appendix E – PocketPC Processes

Platform: PocketPC
Version: 4.20.1081
CPU: StrongARM

Process Memory Dump
filesys.exe

Virtual Memory Dump
==========
Legend
C - Committed
R - Reserved
F - Free

RO - Read Only
RW - Read Write
WC - Write Copy
X - Execute
XR - Execute Read
XRW - Execute Read Write
XWC - Execute Write Copy
G - Guard
NA - No Access
NC - No Cache

04000000 10000 F NA
04010000 1000 R NA Image
04011000 30000 C RO Image
04041000 7000 C RW Image
04048000 3000 C RO Image
0404B000 5000 F NA
04050000 9000 R NA Private
04059000 1000 C RW Private
0405A000 1000 R NA Private
0405B000 1000 C RW Private
0405C000 1000 R NA Private
0405D000 3000 C RW Private
04060000 2F000 C RW Private
0408F000 1000 F NA
04090000 D000 R NA Private
0409D000 3000 C RW Private
040A0000 1000 C RW, NC Private
040A1000 F000 F NA
040B0000 1000 C RW, NC Private
040B1000 F000 F NA
040C0000 F000 R NA Private
040CF000 1000 C RW Private
040D0000 1E000 C RW Private
040EE000 11000 R NA Private
040FF000 1000 F NA
04100000 4000 C RW Private
04104000 C000 F NA
04110000 4000 C RW Private
04114000 C000 F NA
04120000 F000 R NA Private
0412F000 1000 C RW Private
04130000 10000 C RW Private
04140000 18D0000 F NA
05A10000 46000 R NA Image
05A56000 2A000 F NA
05A80000 1C000 R NA Image
05A9C000 B000 C RW Image
05AA7000 7000 R NA Image
05AAE000 1000 C RW Image
05AAF000 1000 R NA Image
05AB0000 1000 C RW Image
05AB1000 4000 R NA Image
05AB5000 11B000 F NA
05BD0000 44000 R NA Image
05C14000 10C000 F NA
05D20000 83000 R NA Image
05DA3000 3D000 F NA
05DE0000 62000 R NA Image
05E42000 8E000 F NA
05ED0000 61000 R NA Image
05F31000 2F000 F NA
05F60000 1000 R NA Image
05F61000 65000 C RO Image
05FC6000 1000 C RW Image
05FC7000 1B000 C RO Image
05FE2000 3000 R NA Image
05FE5000 B000 F NA
05FF0000 8000 R NA Image
05FF8000 2000 C RW Image
05FFA000 1000 R NA Image
05FFB000 1000 C RW Image
05FFC000 1000 R NA Image
05FFD000 1000 C RW Image
05FFE000 1000 R NA Image
05FFF000 1000 C RW Image

Process Heaps Dump

Default Heap
==========
04060050 40 Fixed
04060090 10 Fixed
040600A0 20 Fixed
040600C0 20 Fixed
040600E0 1A0 Fixed
04060280 20 Fixed
040602A0 20 Fixed
040602C0 20 Fixed
040602E0 200 Fixed
040604E0 200 Fixed
040606E0 20 Fixed

04060700 10 Free
04060710 20 Fixed
04060730 30 Fixed
04060760 20 Fixed
04060780 20 Fixed
040607A0 20 Fixed
040607C0 20 Fixed
040607E0 10 Free
040607F0 20 Fixed
04060810 30 Fixed
04060840 250 Fixed
04060A90 20 Fixed
04060AB0 20 Fixed
04060AD0 30 Fixed
04060B00 9E0 Fixed
040614E0 240 Fixed
04061720 20 Fixed
04061740 10 Fixed
04061750 20 Fixed
04061770 20 Fixed
04061790 200 Fixed
04061990 200 Fixed
04061B90 200 Fixed
04061D90 200 Fixed
04061F90 200 Fixed
04062190 200 Fixed
04062390 200 Fixed
04062590 200 Fixed
04062790 200 Fixed
04062990 200 Fixed
04062B90 200 Fixed
04062D90 200 Fixed
04062F90 200 Fixed
04063190 200 Fixed
04063390 200 Fixed
04063590 200 Fixed
04063790 200 Fixed
04063990 200 Fixed
04063B90 200 Fixed
04063D90 200 Fixed
04063F90 200 Fixed
04064190 200 Fixed
04064390 200 Fixed
04064590 200 Fixed
04064790 200 Fixed
04064990 200 Fixed
04064B90 200 Fixed
04064D90 200 Fixed
04064F90 200 Fixed
04065190 200 Fixed
04065390 200 Fixed
04065590 200 Fixed
04065790 200 Fixed
04065990 200 Fixed
04065B90 200 Fixed
04065D90 200 Fixed
04065F90 200 Fixed
04066190 200 Fixed
04066390 200 Fixed
04066590 200 Fixed
04066790 200 Fixed
04066990 200 Fixed
04066B90 200 Fixed
04066D90 200 Fixed
04066F90 200 Fixed
04067190 200 Fixed
04067390 200 Fixed
04067590 200 Fixed
04067790 200 Fixed
04067990 200 Fixed
04067B90 200 Fixed
04067D90 200 Fixed
04067F90 200 Fixed
04068190 200 Fixed
04068390 200 Fixed
04068590 200 Fixed
04068790 200 Fixed
04068990 200 Fixed
04068B90 200 Fixed
04068D90 200 Fixed
04068F90 200 Fixed
04069190 200 Fixed
04069390 200 Fixed
04069590 200 Fixed
04069790 200 Fixed
04069990 200 Fixed
04069B90 200 Fixed
04069D90 200 Fixed
04069F90 200 Fixed
0406A190 200 Fixed
0406A390 200 Fixed
0406A590 200 Fixed
0406A790 200 Fixed
0406A990 200 Fixed
0406AB90 200 Fixed
0406AD90 200 Fixed
0406AF90 200 Fixed
0406B190 200 Fixed
0406B390 200 Fixed
0406B590 200 Fixed
0406B790 200 Fixed
0406B990 200 Fixed
0406BB90 200 Fixed
0406BD90 200 Fixed
0406BF90 200 Fixed
0406C190 200 Fixed
0406C390 200 Fixed
0406C590 200 Fixed
0406C790 200 Fixed
0406C990 200 Fixed
0406CB90 200 Fixed
0406CD90 200 Fixed

0406CF90 200 Fixed
0406D190 200 Fixed
0406D390 200 Fixed
0406D590 200 Fixed
0406D790 200 Fixed
0406D990 200 Fixed
0406DB90 200 Fixed
0406DD90 200 Fixed
0406DF90 200 Fixed
0406E190 200 Fixed
0406E390 200 Fixed
0406E590 200 Fixed
0406E790 200 Fixed
0406E990 200 Fixed
0406EB90 200 Fixed
0406ED90 200 Fixed
0406EF90 200 Fixed
0406F190 200 Fixed
0406F390 200 Fixed
0406F590 200 Fixed
0406F790 200 Fixed
0406F990 200 Fixed
0406FB90 200 Fixed
0406FD90 200 Fixed
0406FF90 200 Fixed
04070190 200 Fixed
04070390 200 Fixed
04070590 200 Fixed
04070790 200 Fixed
04070990 200 Fixed
04070B90 200 Fixed
04070D90 200 Fixed
04070F90 200 Fixed
04071190 200 Fixed
04071390 200 Fixed
04071590 200 Fixed
04071790 200 Fixed
04071990 200 Fixed
04071B90 200 Fixed
04071D90 200 Fixed
04071F90 200 Fixed
04072190 200 Fixed
04072390 200 Fixed
04072590 200 Fixed
04072790 20 Fixed
040727B0 20 Fixed
040727D0 210 Fixed
040729E0 20 Fixed
04072A00 60 Fixed
04072A60 60 Fixed
04072AC0 20 Fixed
04072AE0 20 Fixed
04072B00 40 Fixed
04072B40 30 Fixed
04072B70 80 Fixed
04072BF0 50 Fixed
04072C40 40 Fixed
04072C80 30 Fixed
04072CB0 20 Fixed
04072CD0 90 Fixed
04072D60 90 Fixed
04072DF0 780 Fixed
04073570 720 Fixed
04073C90 250 Fixed
04073EE0 1040 Fixed
04074F20 50 Fixed
04074F70 30 Fixed
04074FA0 10 Free
04074FB0 350 Fixed
04075300 80 Fixed
04075380 50 Fixed
040753D0 40 Fixed
04075410 20 Fixed
04075430 10 Free
04075440 40 Fixed
04075480 20 Fixed
040754A0 30 Fixed
040754D0 40 Fixed
04075510 20 Fixed
04075530 80 Fixed
040755B0 70 Fixed
04075620 40 Fixed
04075660 20 Fixed
04075680 80 Fixed
04075700 50 Fixed
04075750 40 Fixed
04075790 20 Fixed
040757B0 80 Fixed
04075830 50 Fixed
04075880 40 Fixed
040758C0 20 Fixed
040758E0 80 Fixed
04075960 50 Fixed
040759B0 40 Fixed
040759F0 20 Fixed
04075A10 80 Fixed
04075A90 50 Fixed
04075AE0 40 Fixed
04075B20 20 Fixed
04075B40 80 Fixed
04075BC0 50 Fixed
04075C10 40 Fixed
04075C50 30 Fixed
04075C80 80 Fixed
04075D00 50 Fixed
04075D50 40 Fixed
04075D90 20 Fixed
04075DB0 80 Fixed
04075E30 50 Fixed
04075E80 40 Fixed
04075EC0 20 Fixed
04075EE0 80 Fixed

04075F60 70 Fixed
04075FD0 80 Fixed
04076050 70 Fixed
040760C0 40 Fixed
04076100 20 Fixed
04076120 80 Fixed
040761A0 80 Fixed
04076220 40 Fixed
04076260 20 Fixed
04076280 80 Fixed
04076300 70 Fixed
04076370 40 Fixed
040763B0 20 Fixed
040763D0 80 Fixed
04076450 50 Fixed
040764A0 40 Fixed
040764E0 20 Fixed
04076500 40 Fixed
04076540 20 Fixed
04076560 40 Fixed
040765A0 30 Fixed
040765D0 40 Fixed
04076610 20 Fixed
04076630 40 Fixed
04076670 30 Fixed
040766A0 40 Fixed
040766E0 30 Fixed
04076710 40 Fixed
04076750 20 Fixed
04076770 40 Fixed
040767B0 30 Fixed
040767E0 40 Fixed
04076820 20 Fixed
04076840 40 Fixed
04076880 20 Fixed
040768A0 80 Fixed
04076920 70 Fixed
04076990 40 Fixed
040769D0 20 Fixed
040769F0 80 Fixed
04076A70 70 Fixed
04076AE0 40 Fixed
04076B20 20 Fixed
04076B40 40 Fixed
04076B80 80 Fixed
04076C00 40 Fixed
04076C40 80 Fixed
04076CC0 40 Fixed
04076D00 30 Fixed
04076D30 40 Fixed
04076D70 20 Fixed
04076D90 80 Fixed
04076E10 70 Fixed
04076E80 40 Fixed
04076EC0 20 Fixed
04076EE0 80 Fixed
04076F60 50 Fixed
04076FB0 40 Fixed
04076FF0 80 Fixed
04077070 50 Fixed
040770C0 40 Fixed
04077100 20 Fixed
04077120 80 Fixed
040771A0 50 Fixed
040771F0 40 Fixed
04077230 20 Fixed
04077250 80 Fixed
040772D0 50 Fixed
04077320 40 Fixed
04077360 20 Fixed
04077380 80 Fixed
04077400 50 Fixed
04077450 40 Fixed
04077490 20 Fixed
040774B0 80 Fixed
04077530 50 Fixed
04077580 40 Fixed
040775C0 20 Fixed
040775E0 80 Fixed
04077660 50 Fixed
040776B0 40 Fixed
040776F0 20 Fixed
04077710 80 Fixed
04077790 50 Fixed
040777E0 40 Fixed
04077820 20 Fixed
04077840 80 Fixed
040778C0 50 Fixed
04077910 40 Fixed
04077950 20 Fixed
04077970 80 Fixed
040779F0 50 Fixed
04077A40 40 Fixed
04077A80 20 Fixed
04077AA0 80 Fixed
04077B20 50 Fixed
04077B70 40 Fixed
04077BB0 30 Fixed
04077BE0 80 Fixed
04077C60 70 Fixed
04077CD0 40 Fixed
04077D10 20 Fixed
04077D30 80 Fixed
04077DB0 70 Fixed
04077E20 40 Fixed
04077E60 30 Fixed
04077E90 80 Fixed
04077F10 70 Fixed
04077F80 40 Fixed
04077FC0 20 Fixed
04077FE0 80 Fixed
04078060 70 Fixed

040780D0 40 Fixed
04078110 30 Fixed
04078140 80 Fixed
040781C0 70 Fixed
04078230 40 Fixed
04078270 30 Fixed
040782A0 80 Fixed
04078320 70 Fixed
04078390 40 Fixed
040783D0 20 Fixed
040783F0 80 Fixed
04078470 50 Fixed
040784C0 40 Fixed
04078500 20 Fixed
04078520 80 Fixed
040785A0 50 Fixed
040785F0 40 Fixed
04078630 20 Fixed
04078650 80 Fixed
040786D0 50 Fixed
04078720 40 Fixed
04078760 20 Fixed
04078780 80 Fixed
04078800 70 Fixed
04078870 40 Fixed
040788B0 30 Fixed
040788E0 80 Fixed
04078960 50 Fixed
040789B0 40 Fixed
040789F0 30 Fixed
04078A20 80 Fixed
04078AA0 50 Fixed
04078AF0 40 Fixed
04078B30 30 Fixed
04078B60 80 Fixed
04078BE0 50 Fixed
04078C30 40 Fixed
04078C70 30 Fixed
04078CA0 80 Fixed
04078D20 50 Fixed
04078D70 40 Fixed
04078DB0 20 Fixed
04078DD0 80 Fixed
04078E50 50 Fixed
04078EA0 40 Fixed
04078EE0 20 Fixed
04078F00 80 Fixed
04078F80 50 Fixed
04078FD0 40 Fixed
04079010 30 Fixed
04079040 80 Fixed
040790C0 50 Fixed
04079110 40 Fixed
04079150 30 Fixed
04079180 80 Fixed
04079200 70 Fixed
04079270 40 Fixed
040792B0 30 Fixed
040792E0 80 Fixed
04079360 80 Fixed
040793E0 40 Fixed
04079420 30 Fixed
04079450 80 Fixed
040794D0 80 Fixed
04079550 40 Fixed
04079590 30 Fixed
040795C0 80 Fixed
04079640 50 Fixed
04079690 40 Fixed
040796D0 30 Fixed
04079700 80 Fixed
04079780 50 Fixed
040797D0 40 Fixed
04079810 30 Fixed
04079840 80 Fixed
040798C0 50 Fixed
04079910 60 Fixed
04079970 40 Fixed
040799B0 60 Fixed
04079A10 40 Fixed
04079A50 60 Fixed
04079AB0 40 Fixed
04079AF0 70 Fixed
04079B60 40 Fixed
04079BA0 70 Fixed
04079C10 40 Fixed
04079C50 70 Fixed
04079CC0 40 Fixed
04079D00 70 Fixed
04079D70 40 Fixed
04079DB0 90 Fixed
04079E40 40 Fixed
04079E80 60 Fixed
04079EE0 40 Fixed
04079F20 30 Fixed
04079F50 40 Fixed
04079F90 30 Fixed
04079FC0 40 Fixed
0407A000 40 Fixed
0407A040 40 Fixed
0407A080 30 Fixed
0407A0B0 40 Fixed
0407A0F0 30 Fixed
0407A120 40 Fixed
0407A160 20 Fixed
0407A180 40 Fixed
0407A1C0 30 Fixed
0407A1F0 40 Fixed
0407A230 20 Fixed
0407A250 40 Fixed
0407A290 20 Fixed
0407A2B0 40 Fixed

0407A2F0 20 Fixed
0407A310 40 Fixed
0407A350 30 Fixed
0407A380 40 Fixed
0407A3C0 20 Fixed
0407A3E0 40 Fixed
0407A420 20 Fixed
0407A440 40 Fixed
0407A480 20 Fixed
0407A4A0 40 Fixed
0407A4E0 20 Fixed
0407A500 40 Fixed
0407A540 30 Fixed
0407A570 40 Fixed
0407A5B0 20 Fixed
0407A5D0 40 Fixed
0407A610 20 Fixed
0407A630 40 Fixed
0407A670 20 Fixed
0407A690 40 Fixed
0407A6D0 20 Fixed
0407A6F0 40 Fixed
0407A730 20 Fixed
0407A750 40 Fixed
0407A790 30 Fixed
0407A7C0 40 Fixed
0407A800 30 Fixed
0407A830 40 Fixed
0407A870 20 Fixed
0407A890 40 Fixed
0407A8D0 20 Fixed
0407A8F0 40 Fixed
0407A930 20 Fixed
0407A950 40 Fixed
0407A990 20 Fixed
0407A9B0 40 Fixed
0407A9F0 20 Fixed
0407AA10 40 Fixed
0407AA50 30 Fixed
0407AA80 40 Fixed
0407AAC0 20 Fixed
0407AAE0 40 Fixed
0407AB20 30 Fixed
0407AB50 40 Fixed
0407AB90 30 Fixed
0407ABC0 40 Fixed
0407AC00 30 Fixed
0407AC30 40 Fixed
0407AC70 30 Fixed
0407ACA0 40 Fixed
0407ACE0 20 Fixed
0407AD00 40 Fixed
0407AD40 20 Fixed
0407AD60 80 Fixed
0407ADE0 70 Fixed
0407AE50 40 Fixed
0407AE90 20 Fixed
0407AEB0 80 Fixed
0407AF30 70 Fixed
0407AFA0 40 Fixed
0407AFE0 20 Fixed
0407B000 40 Fixed
0407B040 20 Fixed
0407B060 80 Fixed
0407B0E0 50 Fixed
0407B130 40 Fixed
0407B170 20 Fixed
0407B190 80 Fixed
0407B210 70 Fixed
0407B280 40 Fixed
0407B2C0 30 Fixed
0407B2F0 80 Fixed
0407B370 70 Fixed
0407B3E0 40 Fixed
0407B420 20 Fixed
0407B440 80 Fixed
0407B4C0 50 Fixed
0407B510 40 Fixed
0407B550 20 Fixed
0407B570 80 Fixed
0407B5F0 50 Fixed
0407B640 40 Fixed
0407B680 20 Fixed
0407B6A0 80 Fixed
0407B720 50 Fixed
0407B770 40 Fixed
0407B7B0 30 Fixed
0407B7E0 80 Fixed
0407B860 50 Fixed
0407B8B0 40 Fixed
0407B8F0 30 Fixed
0407B920 80 Fixed
0407B9A0 50 Fixed
0407B9F0 40 Fixed
0407BA30 20 Fixed
0407BA50 80 Fixed
0407BAD0 50 Fixed
0407BB20 40 Fixed
0407BB60 20 Fixed
0407BB80 80 Fixed
0407BC00 50 Fixed
0407BC50 40 Fixed
0407BC90 30 Fixed
0407BCC0 80 Fixed
0407BD40 50 Fixed
0407BD90 40 Fixed
0407BDD0 20 Fixed
0407BDF0 80 Fixed
0407BE70 50 Fixed
0407BEC0 40 Fixed
0407BF00 20 Fixed
0407BF20 80 Fixed

0407BFA0 50 Fixed
0407BFF0 40 Fixed
0407C030 30 Fixed
0407C060 80 Fixed
0407C0E0 50 Fixed
0407C130 40 Fixed
0407C170 20 Fixed
0407C190 80 Fixed
0407C210 50 Fixed
0407C260 40 Fixed
0407C2A0 20 Fixed
0407C2C0 80 Fixed
0407C340 50 Fixed
0407C390 40 Fixed
0407C3D0 20 Fixed
0407C3F0 80 Fixed
0407C470 70 Fixed
0407C4E0 40 Fixed
0407C520 20 Fixed
0407C540 80 Fixed
0407C5C0 50 Fixed
0407C610 40 Fixed
0407C650 30 Fixed
0407C680 80 Fixed
0407C700 70 Fixed
0407C770 40 Fixed
0407C7B0 20 Fixed
0407C7D0 80 Fixed
0407C850 70 Fixed
0407C8C0 40 Fixed
0407C900 20 Fixed
0407C920 80 Fixed
0407C9A0 70 Fixed
0407CA10 40 Fixed
0407CA50 30 Fixed
0407CA80 80 Fixed
0407CB00 70 Fixed
0407CB70 40 Fixed
0407CBB0 30 Fixed
0407CBE0 80 Fixed
0407CC60 70 Fixed
0407CCD0 40 Fixed
0407CD10 20 Fixed
0407CD30 80 Fixed
0407CDB0 70 Fixed
0407CE20 40 Fixed
0407CE60 20 Fixed
0407CE80 80 Fixed
0407CF00 50 Fixed
0407CF50 40 Fixed
0407CF90 20 Fixed
0407CFB0 80 Fixed
0407D030 50 Fixed
0407D080 40 Fixed
0407D0C0 20 Fixed
0407D0E0 80 Fixed
0407D160 70 Fixed
0407D1D0 40 Fixed
0407D210 30 Fixed
0407D240 80 Fixed
0407D2C0 70 Fixed
0407D330 40 Fixed
0407D370 30 Fixed
0407D3A0 80 Fixed
0407D420 70 Fixed
0407D490 40 Fixed
0407D4D0 20 Fixed
0407D4F0 80 Fixed
0407D570 50 Fixed
0407D5C0 40 Fixed
0407D600 30 Fixed
0407D630 80 Fixed
0407D6B0 50 Fixed
0407D700 40 Fixed
0407D740 30 Fixed
0407D770 80 Fixed
0407D7F0 50 Fixed
0407D840 40 Fixed
0407D880 20 Fixed
0407D8A0 80 Fixed
0407D920 50 Fixed
0407D970 40 Fixed
0407D9B0 30 Fixed
0407D9E0 80 Fixed
0407DA60 50 Fixed
0407DAB0 40 Fixed
0407DAF0 20 Fixed
0407DB10 80 Fixed
0407DB90 70 Fixed
0407DC00 40 Fixed
0407DC40 20 Fixed
0407DC60 80 Fixed
0407DCE0 70 Fixed
0407DD50 40 Fixed
0407DD90 20 Fixed
0407DDB0 80 Fixed
0407DE30 70 Fixed
0407DEA0 40 Fixed
0407DEE0 20 Fixed
0407DF00 80 Fixed
0407DF80 70 Fixed
0407DFF0 40 Fixed
0407E030 20 Fixed
0407E050 80 Fixed
0407E0D0 70 Fixed
0407E140 40 Fixed
0407E180 30 Fixed
0407E1B0 80 Fixed
0407E230 50 Fixed
0407E280 40 Fixed
0407E2C0 20 Fixed
0407E2E0 80 Fixed

0407E360 50 Fixed
0407E3B0 40 Fixed
0407E3F0 20 Fixed
0407E410 80 Fixed
0407E490 50 Fixed
0407E4E0 40 Fixed
0407E520 20 Fixed
0407E540 80 Fixed
0407E5C0 50 Fixed
0407E610 40 Fixed
0407E650 30 Fixed
0407E680 80 Fixed
0407E700 50 Fixed
0407E750 40 Fixed
0407E790 30 Fixed
0407E7C0 80 Fixed
0407E840 50 Fixed
0407E890 40 Fixed
0407E8D0 20 Fixed
0407E8F0 80 Fixed
0407E970 70 Fixed
0407E9E0 40 Fixed
0407EA20 30 Fixed
0407EA50 80 Fixed
0407EAD0 70 Fixed
0407EB40 40 Fixed
0407EB80 20 Fixed
0407EBA0 80 Fixed
0407EC20 70 Fixed
0407EC90 40 Fixed
0407ECD0 30 Fixed
0407ED00 80 Fixed
0407ED80 70 Fixed
0407EDF0 40 Fixed
0407EE30 20 Fixed
0407EE50 80 Fixed
0407EED0 70 Fixed
0407EF40 40 Fixed
0407EF80 20 Fixed
0407EFA0 80 Fixed
0407F020 50 Fixed
0407F070 40 Fixed
0407F0B0 20 Fixed
0407F0D0 80 Fixed
0407F150 70 Fixed
0407F1C0 40 Fixed
0407F200 30 Fixed
0407F230 80 Fixed
0407F2B0 50 Fixed
0407F300 40 Fixed
0407F340 20 Fixed
0407F360 80 Fixed
0407F3E0 50 Fixed
0407F430 40 Fixed
0407F470 20 Fixed
0407F490 80 Fixed
0407F510 70 Fixed
0407F580 40 Fixed
0407F5C0 30 Fixed
0407F5F0 80 Fixed
0407F670 50 Fixed
0407F6C0 40 Fixed
0407F700 20 Fixed
0407F720 80 Fixed
0407F7A0 50 Fixed
0407F7F0 40 Fixed
0407F830 20 Fixed
0407F850 80 Fixed
0407F8D0 50 Fixed
0407F920 40 Fixed
0407F960 20 Fixed
0407F980 80 Fixed
0407FA00 50 Fixed
0407FA50 40 Fixed
0407FA90 20 Fixed
0407FAB0 80 Fixed
0407FB30 50 Fixed
0407FB80 40 Fixed
0407FBC0 20 Fixed
0407FBE0 80 Fixed
0407FC60 70 Fixed
0407FCD0 40 Fixed
0407FD10 20 Fixed
0407FD30 80 Fixed
0407FDB0 70 Fixed
0407FE20 40 Fixed
0407FE60 30 Fixed
0407FE90 80 Fixed
0407FF10 50 Fixed
0407FF60 40 Fixed
0407FFA0 20 Fixed
0407FFC0 80 Fixed
04080040 70 Fixed
040800B0 40 Fixed
040800F0 20 Fixed
04080110 80 Fixed
04080190 70 Fixed
04080200 40 Fixed
04080240 20 Fixed
04080260 80 Fixed
040802E0 70 Fixed
04080350 40 Fixed
04080390 30 Fixed
040803C0 80 Fixed
04080440 50 Fixed
04080490 40 Fixed
040804D0 20 Fixed
040804F0 80 Fixed
04080570 50 Fixed
040805C0 40 Fixed
04080600 30 Fixed
04080630 80 Fixed

040806B0 50 Fixed
04080700 40 Fixed
04080740 30 Fixed
04080770 80 Fixed
040807F0 70 Fixed
04080860 40 Fixed
040808A0 20 Fixed
040808C0 80 Fixed
04080940 50 Fixed
04080990 40 Fixed
040809D0 20 Fixed
040809F0 80 Fixed
04080A70 70 Fixed
04080AE0 40 Fixed
04080B20 20 Fixed
04080B40 80 Fixed
04080BC0 70 Fixed
04080C30 40 Fixed
04080C70 30 Fixed
04080CA0 80 Fixed
04080D20 50 Fixed
04080D70 40 Fixed
04080DB0 20 Fixed
04080DD0 80 Fixed
04080E50 50 Fixed
04080EA0 40 Fixed
04080EE0 30 Fixed
04080F10 80 Fixed
04080F90 50 Fixed
04080FE0 40 Fixed
04081020 30 Fixed
04081050 80 Fixed
040810D0 80 Fixed
04081150 40 Fixed
04081190 20 Fixed
040811B0 80 Fixed
04081230 50 Fixed
04081280 40 Fixed
040812C0 20 Fixed
040812E0 80 Fixed
04081360 70 Fixed
040813D0 40 Fixed
04081410 20 Fixed
04081430 80 Fixed
040814B0 70 Fixed
04081520 40 Fixed
04081560 30 Fixed
04081590 80 Fixed
04081610 50 Fixed
04081660 40 Fixed
040816A0 20 Fixed
040816C0 80 Fixed
04081740 50 Fixed
04081790 40 Fixed
040817D0 20 Fixed
040817F0 80 Fixed
04081870 70 Fixed
040818E0 40 Fixed
04081920 30 Fixed
04081950 80 Fixed
040819D0 50 Fixed
04081A20 40 Fixed
04081A60 20 Fixed
04081A80 80 Fixed
04081B00 50 Fixed
04081B50 10 Free
04081B60 40 Fixed
04081BA0 20 Fixed
04081BC0 40 Fixed
04081C00 30 Fixed
04081C30 40 Fixed
04081C70 30 Fixed
04081CA0 40 Fixed
04081CE0 20 Fixed
04081D00 40 Fixed
04081D40 20 Fixed
04081D60 40 Fixed
04081DA0 20 Fixed
04081DC0 40 Fixed
04081E00 30 Fixed
04081E30 40 Fixed
04081E70 20 Fixed
04081E90 20 Fixed
04081EB0 40 Fixed
04081EF0 20 Fixed
04081F10 80 Fixed
04081F90 70 Fixed
04082000 40 Fixed
04082040 30 Fixed
04082070 80 Fixed
040820F0 50 Fixed
04082140 40 Fixed
04082180 20 Fixed
040821A0 80 Fixed
04082220 50 Fixed
04082270 40 Fixed
040822B0 30 Fixed
040822E0 80 Fixed
04082360 50 Fixed
040823B0 40 Fixed
040823F0 20 Fixed
04082410 80 Fixed
04082490 50 Fixed
040824E0 40 Fixed
04082520 20 Fixed
04082540 40 Fixed
04082580 20 Fixed
040825A0 10 Free
040825B0 30 Fixed
040825E0 40 Fixed
04082620 30 Fixed
04082650 40 Fixed

04082690	20 Fixed
040826B0	40 Fixed
040826F0	20 Fixed
04082710	40 Fixed
04082750	20 Fixed
04082770	40 Fixed
040827B0	20 Fixed
040827D0	40 Fixed
04082810	20 Fixed
04082830	40 Fixed
04082870	20 Fixed
04082890	40 Fixed
040828D0	20 Fixed
040828F0	40 Fixed
04082930	20 Fixed
04082950	40 Fixed
04082990	20 Fixed
040829B0	40 Fixed
040829F0	20 Fixed
04082A10	40 Fixed
04082A50	20 Fixed
04082A70	40 Fixed
04082AB0	20 Fixed
04082AD0	40 Fixed
04082B10	20 Fixed
04082B30	40 Fixed
04082B70	30 Fixed
04082BA0	40 Fixed
04082BE0	20 Fixed
04082C00	40 Fixed
04082C40	20 Fixed
04082C60	40 Fixed
04082CA0	20 Fixed
04082CC0	40 Fixed
04082D00	20 Fixed
04082D20	40 Fixed
04082D60	20 Fixed
04082D80	40 Fixed
04082DC0	20 Fixed
04082DE0	40 Fixed
04082E20	30 Fixed
04082E50	40 Fixed
04082E90	30 Fixed
04082EC0	40 Fixed
04082F00	20 Fixed
04082F20	40 Fixed
04082F60	30 Fixed
04082F90	40 Fixed
04082FD0	30 Fixed
04083000	40 Fixed
04083040	20 Fixed
04083060	40 Fixed
040830A0	20 Fixed
040830C0	40 Fixed
04083100	20 Fixed
04083120	40 Fixed
04083160	20 Fixed
04083180	40 Fixed
040831C0	20 Fixed
040831E0	40 Fixed
04083220	20 Fixed
04083240	40 Fixed
04083280	20 Fixed
040832A0	40 Fixed
040832E0	20 Fixed
04083300	40 Fixed
04083340	20 Fixed
04083360	40 Fixed
040833A0	20 Fixed
040833C0	40 Fixed
04083400	20 Fixed
04083420	40 Fixed
04083460	20 Fixed
04083480	40 Fixed
040834C0	20 Fixed
040834E0	40 Fixed
04083520	20 Fixed
04083540	40 Fixed
04083580	20 Fixed
040835A0	40 Fixed
040835E0	30 Fixed
04083610	40 Fixed
04083650	20 Fixed
04083670	40 Fixed
040836B0	20 Fixed
040836D0	40 Fixed
04083710	30 Fixed
04083740	40 Fixed
04083780	20 Fixed
040837A0	40 Fixed
040837E0	30 Fixed
04083810	40 Fixed
04083850	20 Fixed
04083870	40 Fixed
040838B0	30 Fixed
040838E0	40 Fixed
04083920	20 Fixed
04083940	40 Fixed
04083980	20 Fixed
040839A0	40 Fixed
040839E0	20 Fixed
04083A00	40 Fixed
04083A40	20 Fixed
04083A60	40 Fixed
04083AA0	30 Fixed
04083AD0	40 Fixed
04083B10	20 Fixed
04083B30	40 Fixed
04083B70	20 Fixed
04083B90	40 Fixed
04083BD0	30 Fixed
04083C00	40 Fixed

04083C40 30 Fixed
04083C70 40 Fixed
04083CB0 20 Fixed
04083CD0 40 Fixed
04083D10 20 Fixed
04083D30 40 Fixed
04083D70 20 Fixed
04083D90 40 Fixed
04083DD0 30 Fixed
04083E00 40 Fixed
04083E40 20 Fixed
04083E60 40 Fixed
04083EA0 20 Fixed
04083EC0 40 Fixed
04083F00 30 Fixed
04083F30 40 Fixed
04083F70 30 Fixed
04083FA0 40 Fixed
04083FE0 30 Fixed
04084010 40 Fixed
04084050 30 Fixed
04084080 40 Fixed
040840C0 30 Fixed
040840F0 40 Fixed
04084130 30 Fixed
04084160 40 Fixed
040841A0 30 Fixed
040841D0 40 Fixed
04084210 30 Fixed
04084240 40 Fixed
04084280 20 Fixed
040842A0 40 Fixed
040842E0 20 Fixed
04084300 40 Fixed
04084340 30 Fixed
04084370 40 Fixed
040843B0 30 Fixed
040843E0 40 Fixed
04084420 30 Fixed
04084450 80 Fixed
040844D0 50 Fixed
04084520 40 Fixed
04084560 30 Fixed
04084590 80 Fixed
04084610 50 Fixed
04084660 40 Fixed
040846A0 20 Fixed
040846C0 80 Fixed
04084740 50 Fixed
04084790 40 Fixed
040847D0 20 Fixed
040847F0 80 Fixed
04084870 50 Fixed
040848C0 40 Fixed
04084900 20 Fixed
04084920 80 Fixed
040849A0 50 Fixed
040849F0 40 Fixed
04084A30 20 Fixed
04084A50 80 Fixed
04084AD0 50 Fixed
04084B20 40 Fixed
04084B60 30 Fixed
04084B90 80 Fixed
04084C10 50 Fixed
04084C60 40 Fixed
04084CA0 20 Fixed
04084CC0 80 Fixed
04084D40 50 Fixed
04084D90 40 Fixed
04084DD0 20 Fixed
04084DF0 80 Fixed
04084E70 50 Fixed
04084EC0 40 Fixed
04084F00 20 Fixed
04084F20 80 Fixed
04084FA0 50 Fixed
04084FF0 80 Fixed
04085070 50 Fixed
040850C0 40 Fixed
04085100 20 Fixed
04085120 80 Fixed
040851A0 50 Fixed
040851F0 40 Fixed
04085230 20 Fixed
04085250 80 Fixed
040852D0 50 Fixed
04085320 40 Fixed
04085360 20 Fixed
04085380 80 Fixed
04085400 50 Fixed
04085450 40 Fixed
04085490 20 Fixed
040854B0 80 Fixed
04085530 50 Fixed
04085580 40 Fixed
040855C0 20 Fixed
040855E0 80 Fixed
04085660 50 Fixed
040856B0 40 Fixed
040856F0 20 Fixed
04085710 80 Fixed
04085790 50 Fixed
040857E0 40 Fixed
04085820 20 Fixed
04085840 80 Fixed
040858C0 50 Fixed
04085910 40 Fixed
04085950 20 Fixed
04085970 80 Fixed
040859F0 50 Fixed
04085A40 40 Fixed

04085A80 20 Fixed
04085AA0 80 Fixed
04085B20 70 Fixed
04085B90 40 Fixed
04085BD0 20 Fixed
04085BF0 80 Fixed
04085C70 70 Fixed
04085CE0 40 Fixed
04085D20 20 Fixed
04085D40 80 Fixed
04085DC0 70 Fixed
04085E30 40 Fixed
04085E70 20 Fixed
04085E90 80 Fixed
04085F10 70 Fixed
04085F80 40 Fixed
04085FC0 20 Fixed
04085FE0 80 Fixed
04086060 70 Fixed
040860D0 40 Fixed
04086110 20 Fixed
04086130 80 Fixed
040861B0 70 Fixed
04086220 40 Fixed
04086260 20 Fixed
04086280 80 Fixed
04086300 70 Fixed
04086370 40 Fixed
040863B0 20 Fixed
040863D0 80 Fixed
04086450 70 Fixed
040864C0 40 Fixed
04086500 20 Fixed
04086520 80 Fixed
040865A0 70 Fixed
04086610 40 Fixed
04086650 20 Fixed
04086670 80 Fixed
040866F0 70 Fixed
04086760 40 Fixed
040867A0 20 Fixed
040867C0 80 Fixed
04086840 70 Fixed
040868B0 40 Fixed
040868F0 30 Fixed
04086920 80 Fixed
040869A0 70 Fixed
04086A10 40 Fixed
04086A50 20 Fixed
04086A70 40 Fixed
04086AB0 30 Fixed
04086AE0 40 Fixed
04086B20 40 Fixed
04086B60 40 Fixed
04086BA0 30 Fixed
04086BD0 40 Fixed
04086C10 30 Fixed
04086C40 40 Fixed
04086C80 30 Fixed
04086CB0 40 Fixed
04086CF0 20 Fixed
04086D10 40 Fixed
04086D50 20 Fixed
04086D70 40 Fixed
04086DB0 20 Fixed
04086DD0 40 Fixed
04086E10 30 Fixed
04086E40 40 Fixed
04086E80 20 Fixed
04086EA0 40 Fixed
04086EE0 30 Fixed
04086F10 40 Fixed
04086F50 30 Fixed
04086F80 40 Fixed
04086FC0 30 Fixed
04086FF0 40 Fixed
04087030 30 Fixed
04087060 40 Fixed
040870A0 30 Fixed
040870D0 20 Fixed
040870F0 40 Fixed
04087130 30 Fixed
04087160 80 Fixed
040871E0 70 Fixed
04087250 40 Fixed
04087290 30 Fixed
040872C0 80 Fixed
04087340 80 Fixed
040873C0 40 Fixed
04087400 20 Fixed
04087420 80 Fixed
040874A0 70 Fixed
04087510 40 Fixed
04087550 30 Fixed
04087580 80 Fixed
04087600 70 Fixed
04087670 40 Fixed
040876B0 20 Fixed
040876D0 80 Fixed
04087750 70 Fixed
040877C0 40 Fixed
04087800 20 Fixed
04087820 80 Fixed
040878A0 50 Fixed
040878F0 40 Fixed
04087930 20 Fixed
04087950 80 Fixed
040879D0 50 Fixed
04087A20 40 Fixed
04087A60 30 Fixed
04087A90 80 Fixed
04087B10 70 Fixed

04087B80 40 Fixed
04087BC0 20 Fixed
04087BE0 80 Fixed
04087C60 50 Fixed
04087CB0 40 Fixed
04087CF0 20 Fixed
04087D10 80 Fixed
04087D90 50 Fixed
04087DE0 40 Fixed
04087E20 20 Fixed
04087E40 80 Fixed
04087EC0 50 Fixed
04087F10 40 Fixed
04087F50 30 Fixed
04087F80 80 Fixed
04088000 70 Fixed
04088070 40 Fixed
040880B0 30 Fixed
040880E0 80 Fixed
04088160 50 Fixed
040881B0 40 Fixed
040881F0 30 Fixed
04088220 80 Fixed
040882A0 70 Fixed
04088310 40 Fixed
04088350 30 Fixed
04088380 80 Fixed
04088400 50 Fixed
04088450 40 Fixed
04088490 20 Fixed
040884B0 80 Fixed
04088530 70 Fixed
040885A0 40 Fixed
040885E0 30 Fixed
04088610 80 Fixed
04088690 50 Fixed
040886E0 40 Fixed
04088720 30 Fixed
04088750 80 Fixed
040887D0 50 Fixed
04088820 40 Fixed
04088860 20 Fixed
04088880 80 Fixed
04088900 70 Fixed
04088970 40 Fixed
040889B0 20 Fixed
040889D0 80 Fixed
04088A50 70 Fixed
04088AC0 40 Fixed
04088B00 20 Fixed
04088B20 80 Fixed
04088BA0 70 Fixed
04088C10 40 Fixed
04088C50 20 Fixed
04088C70 80 Fixed
04088CF0 70 Fixed
04088D60 40 Fixed
04088DA0 20 Fixed
04088DC0 80 Fixed
04088E40 70 Fixed
04088EB0 40 Fixed
04088EF0 20 Fixed
04088F10 80 Fixed
04088F90 70 Fixed
04089000 40 Fixed
04089040 20 Fixed
04089060 80 Fixed
040890E0 70 Fixed
04089150 40 Fixed
04089190 20 Fixed
040891B0 80 Fixed
04089230 70 Fixed
040892A0 40 Fixed
040892E0 20 Fixed
04089300 80 Fixed
04089380 70 Fixed
040893F0 40 Fixed
04089430 30 Fixed
04089460 80 Fixed
040894E0 50 Fixed
04089530 40 Fixed
04089570 30 Fixed
040895A0 80 Fixed
04089620 50 Fixed
04089670 40 Fixed
040896B0 30 Fixed
040896E0 80 Fixed
04089760 E0 Fixed
04089840 40 Fixed
04089880 30 Fixed
040898B0 80 Fixed
04089930 50 Fixed
04089980 40 Fixed
040899C0 20 Fixed
040899E0 80 Fixed
04089A60 50 Fixed
04089AB0 40 Fixed
04089AF0 30 Fixed
04089B20 80 Fixed
04089BA0 70 Fixed
04089C10 40 Fixed
04089C50 20 Fixed
04089C70 80 Fixed
04089CF0 50 Fixed
04089D40 40 Fixed
04089D80 30 Fixed
04089DB0 80 Fixed
04089E30 70 Fixed
04089EA0 40 Fixed
04089EE0 20 Fixed
04089F00 80 Fixed
04089F80 70 Fixed

04089FF0 40 Fixed
0408A030 20 Fixed
0408A050 80 Fixed
0408A0D0 70 Fixed
0408A140 40 Fixed
0408A180 30 Fixed
0408A1B0 80 Fixed
0408A230 50 Fixed
0408A280 40 Fixed
0408A2C0 20 Fixed
0408A2E0 80 Fixed
0408A360 70 Fixed
0408A3D0 40 Fixed
0408A410 30 Fixed
0408A440 80 Fixed
0408A4C0 70 Fixed
0408A530 40 Fixed
0408A570 30 Fixed
0408A5A0 80 Fixed
0408A620 80 Fixed
0408A6A0 40 Fixed
0408A6E0 30 Fixed
0408A710 80 Fixed
0408A790 80 Fixed
0408A810 40 Fixed
0408A850 30 Fixed
0408A880 80 Fixed
0408A900 A0 Fixed
0408A9A0 40 Fixed
0408A9E0 40 Fixed
0408AA20 40 Fixed
0408AA60 30 Fixed
0408AA90 40 Fixed
0408AAD0 30 Fixed
0408AB00 40 Fixed
0408AB40 30 Fixed
0408AB70 40 Fixed
0408ABB0 30 Fixed
0408ABE0 40 Fixed
0408AC20 30 Fixed
0408AC50 40 Fixed
0408AC90 30 Fixed
0408ACC0 40 Fixed
0408AD00 30 Fixed
0408AD30 40 Fixed
0408AD70 30 Fixed
0408ADA0 40 Fixed
0408ADE0 30 Fixed
0408AE10 40 Fixed
0408AE50 30 Fixed
0408AE80 40 Fixed
0408AEC0 30 Fixed
0408AEF0 40 Fixed
0408AF30 30 Fixed
0408AF60 40 Fixed
0408AFA0 30 Fixed
0408AFD0 40 Fixed
0408B010 30 Fixed
0408B040 40 Fixed
0408B080 20 Fixed
0408B0A0 30 Fixed
0408B0D0 40 Fixed
0408B110 20 Fixed
0408B130 80 Fixed
0408B1B0 50 Fixed
0408B200 40 Fixed
0408B240 30 Fixed
0408B270 80 Fixed
0408B2F0 50 Fixed
0408B340 40 Fixed
0408B380 30 Fixed
0408B3B0 40 Fixed
0408B3F0 20 Fixed
0408B410 40 Fixed
0408B450 30 Fixed
0408B480 40 Fixed
0408B4C0 20 Fixed
0408B4E0 40 Fixed
0408B520 30 Fixed
0408B550 40 Fixed
0408B590 30 Fixed
0408B5C0 40 Fixed
0408B600 20 Fixed
0408B620 40 Fixed
0408B660 20 Fixed
0408B680 40 Fixed
0408B6C0 30 Fixed
0408B6F0 40 Fixed
0408B730 20 Fixed
0408B750 40 Fixed
0408B790 30 Fixed
0408B7C0 40 Fixed
0408B800 30 Fixed
0408B830 40 Fixed
0408B870 20 Fixed
0408B890 40 Fixed
0408B8D0 30 Fixed
0408B900 40 Fixed
0408B940 30 Fixed
0408B970 40 Fixed
0408B9B0 30 Fixed
0408B9E0 40 Fixed
0408BA20 20 Fixed
0408BA40 40 Fixed
0408BA80 20 Fixed
0408BAA0 40 Fixed
0408BAE0 20 Fixed
0408BB00 40 Fixed
0408BB40 30 Fixed
0408BB70 40 Fixed
0408BBB0 20 Fixed
0408BBD0 40 Fixed

0408BC10 20 Fixed
0408BC30 40 Fixed
0408BC70 30 Fixed
0408BCA0 40 Fixed
0408BCE0 20 Fixed
0408BD00 40 Fixed
0408BD40 30 Fixed
0408BD70 40 Fixed
0408BDB0 20 Fixed
0408BDD0 40 Fixed
0408BE10 20 Fixed
0408BE30 40 Fixed
0408BE70 30 Fixed
0408BEA0 40 Fixed
0408BEE0 30 Fixed
0408BF10 40 Fixed
0408BF50 30 Fixed
0408BF80 40 Fixed
0408BFC0 30 Fixed
0408BFF0 40 Fixed
0408C030 20 Fixed
0408C050 40 Fixed
0408C090 30 Fixed
0408C0C0 40 Fixed
0408C100 30 Fixed
0408C130 40 Fixed
0408C170 20 Fixed
0408C190 40 Fixed
0408C1D0 20 Fixed
0408C1F0 40 Fixed
0408C230 20 Fixed
0408C250 40 Fixed
0408C290 30 Fixed
0408C2C0 40 Fixed
0408C300 20 Fixed
0408C320 40 Fixed
0408C360 30 Fixed
0408C390 40 Fixed
0408C3D0 20 Fixed
0408C3F0 40 Fixed
0408C430 20 Fixed
0408C450 40 Fixed
0408C490 20 Fixed
0408C4B0 40 Fixed
0408C4F0 30 Fixed
0408C520 40 Fixed
0408C560 20 Fixed
0408C580 40 Fixed
0408C5C0 30 Fixed
0408C5F0 40 Fixed
0408C630 30 Fixed
0408C660 40 Fixed
0408C6A0 20 Fixed
0408C6C0 40 Fixed
0408C700 20 Fixed
0408C720 40 Fixed
0408C760 20 Fixed
0408C780 40 Fixed
0408C7C0 20 Fixed
0408C7E0 40 Fixed
0408C820 20 Fixed
0408C840 40 Fixed
0408C880 20 Fixed
0408C8A0 40 Fixed
0408C8E0 30 Fixed
0408C910 40 Fixed
0408C950 30 Fixed
0408C980 40 Fixed
0408C9C0 30 Fixed
0408C9F0 40 Fixed
0408CA30 20 Fixed
0408CA50 40 Fixed
0408CA90 20 Fixed
0408CAB0 40 Fixed
0408CAF0 20 Fixed
0408CB10 40 Fixed
0408CB50 20 Fixed
0408CB70 40 Fixed
0408CBB0 20 Fixed
0408CBD0 40 Fixed
0408CC10 20 Fixed
0408CC30 40 Fixed
0408CC70 30 Fixed
0408CCA0 40 Fixed
0408CCE0 20 Fixed
0408CD00 40 Fixed
0408CD40 20 Fixed
0408CD60 40 Fixed
0408CDA0 20 Fixed
0408CDC0 40 Fixed
0408CE00 20 Fixed
0408CE20 40 Fixed
0408CE60 20 Fixed
0408CE80 40 Fixed
0408CEC0 20 Fixed
0408CEE0 40 Fixed
0408CF20 20 Fixed
0408CF40 40 Fixed
0408CF80 20 Fixed
0408CFA0 40 Fixed
0408CFE0 30 Fixed
0408D010 40 Fixed
0408D050 40 Fixed
0408D090 40 Fixed
0408D0D0 20 Fixed
0408D0F0 40 Fixed
0408D130 20 Fixed
0408D150 40 Fixed
0408D190 20 Fixed
0408D1B0 40 Fixed
0408D1F0 20 Fixed
0408D210 40 Fixed

0408D250 20 Fixed
0408D270 40 Fixed
0408D2B0 30 Fixed
0408D2E0 40 Fixed
0408D320 20 Fixed
0408D340 40 Fixed
0408D380 30 Fixed
0408D3B0 40 Fixed
0408D3F0 30 Fixed
0408D420 40 Fixed
0408D460 20 Fixed
0408D480 40 Fixed
0408D4C0 30 Fixed
0408D4F0 80 Fixed
0408D570 50 Fixed
0408D5C0 40 Fixed
0408D600 20 Fixed
0408D620 80 Fixed
0408D6A0 50 Fixed
0408D6F0 40 Fixed
0408D730 20 Fixed
0408D750 80 Fixed
0408D7D0 50 Fixed
0408D820 40 Fixed
0408D860 20 Fixed
0408D880 80 Fixed
0408D900 50 Fixed
0408D950 40 Fixed
0408D990 20 Fixed
0408D9B0 80 Fixed
0408DA30 50 Fixed
0408DA80 40 Fixed
0408DAC0 20 Fixed
0408DAE0 80 Fixed
0408DB60 50 Fixed
0408DBB0 40 Fixed
0408DBF0 20 Fixed
0408DC10 80 Fixed
0408DC90 50 Fixed
0408DCE0 40 Fixed
0408DD20 30 Fixed
0408DD50 80 Fixed
0408DDD0 50 Fixed
0408DE20 40 Fixed
0408DE60 30 Fixed
0408DE90 80 Fixed
0408DF10 70 Fixed
0408DF80 40 Fixed
0408DFC0 30 Fixed
0408DFF0 80 Fixed
0408E070 70 Fixed
0408E0E0 40 Fixed
0408E120 40 Fixed
0408E160 80 Fixed
0408E1E0 70 Fixed
0408E250 40 Fixed
0408E290 20 Fixed
0408E2B0 80 Fixed
0408E330 70 Fixed
0408E3A0 40 Fixed
0408E3E0 30 Fixed
0408E410 80 Fixed
0408E490 70 Fixed
0408E500 40 Fixed
0408E540 30 Fixed
0408E570 80 Fixed
0408E5F0 70 Fixed
0408E660 40 Fixed
0408E6A0 30 Fixed
0408E6D0 80 Fixed
0408E750 70 Fixed
0408E7C0 40 Fixed
0408E800 30 Fixed
0408E830 80 Fixed
0408E8B0 70 Fixed
0408E920 40 Fixed
0408E960 30 Fixed
0408E990 80 Fixed
0408EA10 70 Fixed
0408EA80 40 Fixed
0408EAC0 20 Fixed
0408EAE0 80 Fixed
0408EB60 70 Fixed
0408EBD0 40 Fixed
0408EC10 30 Fixed
0408EC40 80 Fixed
0408ECC0 70 Fixed
0408ED30 40 Fixed
0408ED70 30 Fixed
0408EDA0 80 Fixed
0408EE20 70 Fixed
0408EE90 40 Fixed
0408EED0 30 Fixed
0408EF00 80 Fixed
0408EF80 50 Fixed
0408EFD0 30 Fixed
040D0020 80 Fixed
040D00A0 50 Fixed
040D00F0 40 Fixed
040D0130 30 Fixed
040D0160 80 Fixed
040D01E0 50 Fixed
040D0230 40 Fixed
040D0270 80 Fixed
040D02F0 70 Fixed
040D0360 40 Fixed
040D03A0 80 Fixed
040D0420 70 Fixed
040D0490 40 Fixed
040D04D0 30 Fixed
040D0500 80 Fixed
040D0580 50 Fixed

040D05D0 40 Fixed
040D0610 30 Fixed
040D0640 80 Fixed
040D06C0 50 Fixed
040D0710 40 Fixed
040D0750 30 Fixed
040D0780 80 Fixed
040D0800 70 Fixed
040D0870 40 Fixed
040D08B0 80 Fixed
040D0930 50 Fixed
040D0980 40 Fixed
040D09C0 30 Fixed
040D09F0 80 Fixed
040D0A70 50 Fixed
040D0AC0 40 Fixed
040D0B00 80 Fixed
040D0B80 70 Fixed
040D0BF0 40 Fixed
040D0C30 80 Fixed
040D0CB0 50 Fixed
040D0D00 40 Fixed
040D0D40 80 Fixed
040D0DC0 50 Fixed
040D0E10 40 Fixed
040D0E50 80 Fixed
040D0ED0 50 Fixed
040D0F20 40 Fixed
040D0F60 30 Fixed
040D0F90 80 Fixed
040D1010 70 Fixed
040D1080 40 Fixed
040D10C0 30 Fixed
040D10F0 80 Fixed
040D1170 50 Fixed
040D11C0 40 Fixed
040D1200 30 Fixed
040D1230 80 Fixed
040D12B0 70 Fixed
040D1320 40 Fixed
040D1360 30 Fixed
040D1390 80 Fixed
040D1410 50 Fixed
040D1460 40 Fixed
040D14A0 30 Fixed
040D14D0 80 Fixed
040D1550 50 Fixed
040D15A0 40 Fixed
040D15E0 30 Fixed
040D1610 80 Fixed
040D1690 70 Fixed
040D1700 40 Fixed
040D1740 30 Fixed
040D1770 80 Fixed
040D17F0 80 Fixed
040D1870 40 Fixed
040D18B0 30 Fixed
040D18E0 80 Fixed
040D1960 80 Fixed
040D19E0 40 Fixed
040D1A20 30 Fixed
040D1A50 80 Fixed
040D1AD0 80 Fixed
040D1B50 40 Fixed
040D1B90 30 Fixed
040D1BC0 80 Fixed
040D1C40 80 Fixed
040D1CC0 40 Fixed
040D1D00 30 Fixed
040D1D30 80 Fixed
040D1DB0 80 Fixed
040D1E30 40 Fixed
040D1E70 30 Fixed
040D1EA0 80 Fixed
040D1F20 80 Fixed
040D1FA0 40 Fixed
040D1FE0 30 Fixed
040D2010 80 Fixed
040D2090 80 Fixed
040D2110 40 Fixed
040D2150 30 Fixed
040D2180 80 Fixed
040D2200 40 Fixed
040D2240 20 Fixed
040D2260 80 Fixed
040D22E0 80 Fixed
040D2360 40 Fixed
040D23A0 20 Fixed
040D23C0 80 Fixed
040D2440 80 Fixed
040D24C0 40 Fixed
040D2500 30 Fixed
040D2530 80 Fixed
040D25B0 80 Fixed
040D2630 40 Fixed
040D2670 30 Fixed
040D26A0 80 Fixed
040D2720 80 Fixed
040D27A0 40 Fixed
040D27E0 30 Fixed
040D2810 80 Fixed
040D2890 80 Fixed
040D2910 40 Fixed
040D2950 30 Fixed
040D2980 80 Fixed
040D2A00 80 Fixed
040D2A80 40 Fixed
040D2AC0 40 Fixed
040D2B00 80 Fixed
040D2B80 80 Fixed
040D2C00 B0 Fixed
040D2CB0 30 Fixed

040D2CE0 20 Fixed
040D2D00 40 Fixed
040D2D40 20 Fixed
040D2D60 780 Fixed
040D34E0 20 Fixed
040D3500 20 Fixed
040D3520 20 Fixed
040D3540 10 Free
040D3550 30 Fixed
040D3580 30 Fixed
040D35B0 20 Fixed
040D35D0 30 Fixed
040D3600 30 Fixed
040D3630 30 Fixed
040D3660 30 Fixed
040D3690 30 Fixed
040D36C0 30 Fixed
040D36F0 30 Fixed
040D3720 30 Fixed
040D3750 30 Fixed
040D3780 30 Fixed
040D37B0 30 Fixed
040D37E0 30 Fixed
040D3810 30 Fixed
040D3840 30 Fixed
040D3870 10 Free
040D3880 40 Fixed
040D38C0 40 Fixed
040D3900 20 Fixed
040D3920 40 Fixed
040D3960 20 Fixed
040D3980 40 Fixed
040D39C0 30 Fixed
040D39F0 40 Fixed
040D3A30 20 Fixed
040D3A50 40 Fixed
040D3A90 30 Fixed
040D3AC0 40 Fixed
040D3B00 20 Fixed
040D3B20 40 Fixed
040D3B60 30 Fixed
040D3B90 40 Fixed
040D3BD0 30 Fixed
040D3C00 40 Fixed
040D3C40 20 Fixed
040D3C60 40 Fixed
040D3CA0 20 Fixed
040D3CC0 40 Fixed
040D3D00 20 Fixed
040D3D20 40 Fixed
040D3D60 30 Fixed
040D3D90 40 Fixed
040D3DD0 30 Fixed
040D3E00 40 Fixed
040D3E40 30 Fixed
040D3E70 40 Fixed
040D3EB0 20 Fixed
040D3ED0 40 Fixed
040D3F10 20 Fixed
040D3F30 40 Fixed
040D3F70 20 Fixed
040D3F90 40 Fixed
040D3FD0 30 Fixed
040D4000 40 Fixed
040D4040 30 Fixed
040D4070 40 Fixed
040D40B0 20 Fixed
040D40D0 40 Fixed
040D4110 20 Fixed
040D4130 40 Fixed
040D4170 30 Fixed
040D41A0 40 Fixed
040D41E0 20 Fixed
040D4200 40 Fixed
040D4240 30 Fixed
040D4270 40 Fixed
040D42B0 20 Fixed
040D42D0 40 Fixed
040D4310 30 Fixed
040D4340 40 Fixed
040D4380 20 Fixed
040D43A0 40 Fixed
040D43E0 20 Fixed
040D4400 40 Fixed
040D4440 20 Fixed
040D4460 40 Fixed
040D44A0 20 Fixed
040D44C0 40 Fixed
040D4500 20 Fixed
040D4520 40 Fixed
040D4560 30 Fixed
040D4590 40 Fixed
040D45D0 30 Fixed
040D4600 40 Fixed
040D4640 20 Fixed
040D4660 40 Fixed
040D46A0 30 Fixed
040D46D0 40 Fixed
040D4710 20 Fixed
040D4730 40 Fixed
040D4770 30 Fixed
040D47A0 40 Fixed
040D47E0 20 Fixed
040D4800 40 Fixed
040D4840 20 Fixed
040D4860 40 Fixed
040D48A0 20 Fixed
040D48C0 40 Fixed
040D4900 20 Fixed
040D4920 40 Fixed
040D4960 20 Fixed
040D4980 40 Fixed

040D49C0 30 Fixed
040D49F0 40 Fixed
040D4A30 30 Fixed
040D4A60 40 Fixed
040D4AA0 30 Fixed
040D4AD0 40 Fixed
040D4B10 30 Fixed
040D4B40 40 Fixed
040D4B80 20 Fixed
040D4BA0 40 Fixed
040D4BE0 30 Fixed
040D4C10 40 Fixed
040D4C50 20 Fixed
040D4C70 40 Fixed
040D4CB0 30 Fixed
040D4CE0 40 Fixed
040D4D20 30 Fixed
040D4D50 40 Fixed
040D4D90 30 Fixed
040D4DC0 40 Fixed
040D4E00 30 Fixed
040D4E30 40 Fixed
040D4E70 30 Fixed
040D4EA0 40 Fixed
040D4EE0 30 Fixed
040D4F10 40 Fixed
040D4F50 30 Fixed
040D4F80 40 Fixed
040D4FC0 30 Fixed
040D4FF0 40 Fixed
040D5030 20 Fixed
040D5050 40 Fixed
040D5090 30 Fixed
040D50C0 40 Fixed
040D5100 20 Fixed
040D5120 40 Fixed
040D5160 30 Fixed
040D5190 40 Fixed
040D51D0 30 Fixed
040D5200 40 Fixed
040D5240 30 Fixed
040D5270 40 Fixed
040D52B0 30 Fixed
040D52E0 40 Fixed
040D5320 30 Fixed
040D5350 40 Fixed
040D5390 30 Fixed
040D53C0 40 Fixed
040D5400 30 Fixed
040D5430 40 Fixed
040D5470 30 Fixed
040D54A0 40 Fixed
040D54E0 30 Fixed
040D5510 40 Fixed
040D5550 30 Fixed
040D5580 40 Fixed
040D55C0 30 Fixed
040D55F0 40 Fixed
040D5630 30 Fixed
040D5660 40 Fixed
040D56A0 40 Fixed
040D56E0 40 Fixed
040D5720 30 Fixed
040D5750 40 Fixed
040D5790 20 Fixed
040D57B0 40 Fixed
040D57F0 30 Fixed
040D5820 40 Fixed
040D5860 20 Fixed
040D5880 40 Fixed
040D58C0 40 Fixed
040D5900 40 Fixed
040D5940 40 Fixed
040D5980 40 Fixed
040D59C0 40 Fixed
040D5A00 40 Fixed
040D5A40 30 Fixed
040D5A70 40 Fixed
040D5AB0 20 Fixed
040D5AD0 40 Fixed
040D5B10 30 Fixed
040D5B40 40 Fixed
040D5B80 20 Fixed
040D5BA0 40 Fixed
040D5BE0 40 Fixed
040D5C20 40 Fixed
040D5C60 30 Fixed
040D5C90 40 Fixed
040D5CD0 30 Fixed
040D5D00 40 Fixed
040D5D40 30 Fixed
040D5D70 40 Fixed
040D5DB0 30 Fixed
040D5DE0 40 Fixed
040D5E20 30 Fixed
040D5E50 40 Fixed
040D5E90 20 Fixed
040D5EB0 40 Fixed
040D5EF0 30 Fixed
040D5F20 40 Fixed
040D5F60 30 Fixed
040D5F90 40 Fixed
040D5FD0 30 Fixed
040D6000 40 Fixed
040D6040 30 Fixed
040D6070 40 Fixed
040D60B0 20 Fixed
040D60D0 40 Fixed
040D6110 20 Fixed
040D6130 720 Fixed
040D6850 250 Fixed
040D6AA0 20 Fixed

040D6AC0 30 Fixed
040D6AF0 30 Fixed
040D6B20 190 Free
040D6CB0 50 Fixed
040D6D00 F0 Fixed
040D6DF0 A0 Fixed
040D6E90 A0 Fixed
040D6F30 90 Fixed
040D6FC0 30 Fixed
040D6FF0 30 Fixed
040D7020 30 Fixed
040D7050 30 Fixed
040D7080 30 Fixed
040D70B0 30 Fixed
040D70E0 30 Fixed
040D7110 30 Fixed
040D7140 30 Fixed
040D7170 30 Fixed
040D71A0 30 Fixed
040D71D0 30 Fixed
040D7200 30 Fixed
040D7230 30 Fixed
040D7260 30 Fixed
040D7290 30 Fixed
040D72C0 30 Fixed
040D72F0 30 Fixed
040D7320 30 Fixed
040D7350 B0 Fixed
040D7400 30 Fixed
040D7430 40 Fixed
040D7470 50 Fixed
040D74C0 30 Free
040D74F0 210 Fixed
040D7700 40 Fixed
040D7740 20 Fixed
040D7760 210 Fixed
040D7970 B0 Fixed
040D7A20 40 Fixed
040D7A60 40 Fixed
040D7AA0 80 Free
040D7B20 30 Fixed
040D7B50 2D0 Fixed
040D7E20 120 Free
040D7F40 720 Fixed
040D8660 4A10 Fixed
040DD070 20 Fixed
040DD090 10 Free
040DD0A0 20 Fixed
040DD0C0 30 Fixed
040DD0F0 20 Fixed
040DD110 20 Fixed
040DD130 10 Free
040DD140 20 Fixed
040DD160 30 Fixed
040DD190 20 Fixed
040DD1B0 20 Fixed
040DD1D0 10 Free
040DD1E0 20 Fixed
040DD200 30 Fixed
040DD230 20 Free
040DD250 20 Fixed
040DD270 20 Fixed
040DD290 20 Fixed
040DD2B0 20 Fixed
040DD2D0 20 Fixed
040DD2F0 20 Fixed
040DD310 410 Fixed
040DD720 410 Fixed
040DDB30 410 Fixed
040DDF40 410 Fixed
040DE350 100 Fixed
040DE450 20 Fixed
040DE470 20 Fixed
040DE490 20 Fixed
040DE4B0 20 Fixed
040DE4D0 20 Fixed
040DE4F0 20 Fixed
040DE510 20 Fixed
040DE530 20 Fixed
040DE550 20 Fixed
040DE570 20 Fixed
040DE590 40 Free
040DE5D0 20 Fixed
040DE5F0 20 Fixed
040DE610 10 Free
040DE620 20 Fixed
040DE640 30 Fixed
040DE670 40 Free
040DE6B0 20 Fixed
040DE6D0 20 Free
040DE6F0 20 Fixed
040DE710 20 Fixed
040DE730 10 Free
040DE740 20 Fixed
040DE760 30 Fixed
040DE790 20 Fixed
040DE7B0 20 Fixed
040DE7D0 10 Free
040DE7E0 20 Fixed
040DE800 30 Fixed
040DE830 20 Fixed
040DE850 20 Fixed
040DE870 10 Free
040DE880 20 Fixed
040DE8A0 30 Fixed
040DE8D0 20 Fixed
040DE8F0 20 Fixed
040DE910 20 Fixed
040DE930 30 Fixed
040DE960 30 Fixed
040DE990 20 Fixed
040DE9B0 10 Free

040DE9C0 20 Fixed
040DE9E0 30 Fixed
040DEA10 20 Fixed
040DEA30 20 Fixed
040DEA50 20 Fixed
040DEA70 10 Free
040DEA80 20 Fixed
040DEAA0 30 Fixed
040DEAD0 20 Fixed
040DEAF0 20 Fixed
040DEB10 10 Free
040DEB20 20 Fixed
040DEB40 30 Fixed
040DEB70 20 Fixed
040DEB90 20 Fixed
040DEBB0 10 Free
040DEBC0 20 Fixed
040DEBE0 30 Fixed
040DEC10 20 Fixed
040DEC30 20 Fixed
040DEC50 20 Fixed
040DEC70 30 Fixed
040DECA0 30 Fixed
040DECD0 20 Fixed
040DECF0 10 Free
040DED00 20 Fixed
040DED20 30 Fixed
040DED50 20 Fixed
040DED70 20 Fixed
040DED90 20 Fixed
040DEDB0 20 Fixed
040DEDD0 20 Fixed
040DEDF0 20 Free
040DEE10 30 Fixed
040DEE40 30 Fixed
040DEE70 30 Fixed
040DEEA0 30 Fixed
040DEED0 20 Fixed
040DEEF0 1A0 Fixed
040DF090 20 Fixed
040DF0B0 10 Free
040DF0C0 20 Fixed
040DF0E0 30 Fixed
040DF110 20 Fixed
040DF130 20 Fixed
040DF150 10 Free
040DF160 20 Fixed
040DF180 30 Fixed
040DF1B0 20 Fixed
040DF1D0 10 Free
040DF1E0 20 Fixed
040DF200 30 Fixed
040DF230 20 Fixed
040DF250 20 Fixed
040DF270 10 Free
040DF280 20 Fixed
040DF2A0 30 Fixed
040DF2D0 1C0 Free
040DF490 20 Fixed
040DF4B0 20 Fixed
040DF4D0 10 Free
040DF4E0 20 Fixed
040DF500 30 Fixed
040DF530 20 Fixed
040DF550 20 Fixed
040DF570 10 Free
040DF580 20 Fixed
040DF5A0 10 Free
040DF5B0 20 Fixed
040DF5D0 20 Free
040DF5F0 20 Fixed
040DF610 D0 Free
040DF6E0 20 Fixed
040DF700 60 Free
040DF760 20 Fixed
040DF780 20 Fixed
040DF7A0 10 Free
040DF7B0 20 Fixed
040DF7D0 20 Fixed
040DF7F0 20 Free
040DF810 20 Fixed
040DF830 30 Fixed
040DF860 30 Fixed
040DF890 210 Fixed
040DFAA0 10 Free
040DFAB0 20 Fixed
040DFAD0 20 Fixed
040DFAF0 10 Free
040DFB00 20 Fixed
040DFB20 30 Fixed
040DFB50 20 Fixed
040DFB70 20 Fixed
040DFB90 20 Free
040DFBB0 30 Fixed
040DFBE0 20 Fixed
040DFC00 20 Fixed
040DFC20 10 Free
040DFC30 20 Fixed
040DFC50 30 Fixed
040DFC80 20 Fixed
040DFCA0 10 Free
040DFCB0 20 Fixed
040DFCD0 20 Fixed
040DFCF0 10 Free
040DFD00 20 Fixed
040DFD20 30 Fixed
040DFD50 110 Fixed
040DFE60 80 Fixed
040DFEE0 80 Fixed
040DFF60 20 Fixed
040DFF80 20 Fixed
040DFFA0 20 Fixed

040DFFC0 20 Free
040DFFE0 30 Fixed
040E0010 30 Fixed
040E0040 20 Fixed
040E0060 20 Fixed
040E0080 10 Free
040E0090 20 Fixed
040E00B0 30 Fixed
040E00E0 20 Fixed
040E0100 20 Fixed
040E0120 10 Free
040E0130 20 Fixed
040E0150 30 Fixed
040E0180 20 Fixed
040E01A0 20 Fixed
040E01C0 20 Free
040E01E0 30 Fixed
040E0210 30 Fixed
040E0240 20 Fixed
040E0260 20 Free
040E0280 150 Fixed
040E03D0 20 Fixed
040E03F0 30 Fixed
040E0420 30 Fixed
040E0450 20 Fixed
040E0470 20 Fixed
040E0490 10 Free
040E04A0 20 Fixed
040E04C0 30 Fixed
040E04F0 20 Fixed
040E0510 20 Fixed
040E0530 10 Free
040E0540 20 Fixed
040E0560 30 Fixed
040E0590 20 Fixed
040E05B0 20 Fixed
040E05D0 10 Free
040E05E0 20 Fixed
040E0600 30 Fixed
040E0630 20 Fixed
040E0650 20 Fixed
040E0670 20 Free
040E0690 30 Fixed
040E06C0 30 Fixed
040E06F0 20 Free
040E0710 20 Fixed
040E0730 20 Fixed
040E0750 10 Free
040E0760 20 Fixed
040E0780 30 Fixed
040E07B0 20 Fixed
040E07D0 20 Fixed
040E07F0 10 Free
040E0800 20 Fixed
040E0820 30 Fixed
040E0850 20 Fixed
040E0870 20 Fixed
040E0890 10 Free
040E08A0 20 Fixed
040E08C0 30 Fixed
040E08F0 20 Fixed
040E0910 20 Fixed
040E0930 20 Fixed
040E0950 10 Free
040E0960 20 Fixed
040E0980 30 Fixed
040E09B0 20 Fixed
040E09D0 20 Fixed
040E09F0 10 Free
040E0A00 20 Fixed
040E0A20 30 Fixed
040E0A50 20 Fixed
040E0A70 20 Fixed
040E0A90 20 Free
040E0AB0 30 Fixed
040E0AE0 30 Fixed
040E0B10 20 Fixed
040E0B30 20 Fixed
040E0B50 10 Free
040E0B60 20 Fixed
040E0B80 30 Fixed
040E0BB0 20 Fixed
040E0BD0 20 Fixed
040E0BF0 10 Free
040E0C00 20 Fixed
040E0C20 30 Fixed
040E0C50 20 Fixed
040E0C70 20 Fixed
040E0C90 10 Free
040E0CA0 20 Fixed
040E0CC0 30 Fixed
040E0CF0 20 Fixed
040E0D10 20 Fixed
040E0D30 10 Free
040E0D40 20 Fixed
040E0D60 30 Fixed
040E0D90 340 Fixed
040E10D0 2A0 Fixed
040E1370 2A0 Fixed
040E1610 2A0 Fixed
040E18B0 2A0 Fixed
040E1B50 20 Fixed
040E1B70 20 Fixed
040E1B90 10 Free
040E1BA0 20 Fixed
040E1BC0 30 Fixed
040E1BF0 20 Fixed
040E1C10 20 Fixed
040E1C30 20 Free
040E1C50 30 Fixed
040E1C80 30 Fixed
040E1CB0 20 Free

040E1CD0 20 Fixed
040E1CF0 20 Fixed
040E1D10 10 Free
040E1D20 20 Fixed
040E1D40 30 Fixed
040E1D70 320 Fixed
040E2090 260 Fixed
040E22F0 260 Fixed
040E2550 260 Fixed
040E27B0 260 Fixed
040E2A10 260 Fixed
040E2C70 260 Fixed
040E2ED0 260 Fixed
040E3130 260 Fixed
040E3390 260 Fixed
040E35F0 260 Fixed
040E3850 260 Fixed
040E3AB0 260 Fixed
040E3D10 260 Fixed
040E3F70 260 Fixed
040E41D0 260 Fixed
040E4430 260 Fixed
040E4690 2C0 Fixed
040E4950 230 Fixed
040E4B80 230 Fixed
040E4DB0 230 Fixed
040E4FE0 230 Fixed
040E5210 1C0 Free
040E53D0 20 Fixed
040E53F0 20 Fixed
040E5410 10 Free
040E5420 20 Fixed
040E5440 30 Fixed
040E5470 C0 Free
040E5530 20 Fixed
040E5550 20 Fixed
040E5570 10 Free
040E5580 20 Fixed
040E55A0 30 Fixed
040E55D0 C0 Free
040E5690 20 Fixed
040E56B0 20 Fixed
040E56D0 20 Free
040E56F0 30 Fixed
040E5720 30 Fixed
040E5750 20 Fixed
040E5770 40 Free
040E57B0 20 Fixed
040E57D0 40 Free
040E5810 20 Fixed
040E5830 20 Fixed
040E5850 20 Free
040E5870 30 Fixed
040E58A0 30 Fixed
040E58D0 60 Free
040E5930 30 Fixed
040E5960 30 Fixed
040E5990 140 Free
040E5AD0 20 Fixed
040E5AF0 20 Fixed
040E5B10 30 Fixed
040E5B40 10 Free
040E5B50 20 Fixed
040E5B70 20 Fixed
040E5B90 10 Free
040E5BA0 20 Fixed
040E5BC0 30 Fixed
040E5BF0 20 Fixed
040E5C10 20 Fixed
040E5C30 10 Free
040E5C40 20 Fixed
040E5C60 30 Fixed
040E5C90 30 Fixed
040E5CC0 20 Fixed
040E5CE0 20 Fixed
040E5D00 20 Free
040E5D20 30 Fixed
040E5D50 20 Fixed
040E5D70 20 Fixed
040E5D90 10 Free
040E5DA0 20 Fixed
040E5DC0 30 Fixed
040E5DF0 20 Fixed
040E5E10 20 Fixed
040E5E30 10 Free
040E5E40 20 Fixed
040E5E60 30 Fixed
040E5E90 E0 Fixed
040E5F70 40 Fixed
040E5FB0 40 Fixed
040E5FF0 40 Fixed
040E6030 40 Fixed
040E6070 50 Fixed
040E60C0 50 Fixed
040E6110 30 Fixed
040E6140 20 Fixed
040E6160 20 Fixed
040E6180 20 Free
040E61A0 20 Fixed
040E61C0 10 Free
040E61D0 20 Fixed
040E61F0 20 Fixed
040E6210 10 Free
040E6220 20 Fixed
040E6240 10 Free
040E6250 20 Fixed
040E6270 20 Fixed
040E6290 10 Free
040E62A0 20 Fixed
040E62C0 30 Fixed
040E62F0 30 Fixed
040E6320 20 Fixed

040E6340	20 Fixed
040E6360	20 Free
040E6380	20 Fixed
040E63A0	20 Fixed
040E63C0	20 Free
040E63E0	30 Fixed
040E6410	30 Fixed
040E6440	30 Fixed
040E6470	30 Fixed
040E64A0	20 Free
040E64C0	220 Fixed
040E66E0	30 Fixed
040E6710	20 Fixed
040E6730	20 Fixed
040E6750	10 Free
040E6760	20 Fixed
040E6780	30 Fixed
040E67B0	20 Fixed
040E67D0	20 Fixed
040E67F0	10 Free
040E6800	20 Fixed
040E6820	30 Fixed
040E6850	20 Fixed
040E6870	20 Fixed
040E6890	20 Free
040E68B0	30 Fixed
040E68E0	30 Fixed
040E6910	20 Fixed
040E6930	20 Free
040E6950	220 Fixed
040E6B70	30 Fixed
040E6BA0	20 Fixed
040E6BC0	20 Free
040E6BE0	30 Fixed
040E6C10	30 Fixed
040E6C40	20 Fixed
040E6C60	20 Fixed
040E6C80	10 Free
040E6C90	20 Fixed
040E6CB0	30 Fixed
040E6CE0	20 Fixed
040E6D00	20 Fixed
040E6D20	10 Free
040E6D30	20 Fixed
040E6D50	30 Fixed
040E6D80	20 Fixed
040E6DA0	20 Fixed
040E6DC0	20 Free
040E6DE0	220 Fixed
040E7000	30 Fixed
040E7030	30 Fixed
040E7060	30 Fixed
040E7090	20 Fixed
040E70B0	20 Fixed
040E70D0	10 Free
040E70E0	20 Fixed
040E7100	30 Fixed
040E7130	20 Fixed
040E7150	20 Fixed
040E7170	20 Free
040E7190	30 Fixed
040E71C0	30 Fixed
040E71F0	20 Fixed
040E7210	20 Fixed
040E7230	10 Free
040E7240	20 Fixed
040E7260	30 Fixed
040E7290	20 Fixed
040E72B0	20 Fixed
040E72D0	20 Free
040E72F0	30 Fixed
040E7320	30 Fixed
040E7350	20 Fixed
040E7370	10 Free
040E7380	20 Fixed
040E73A0	30 Fixed
040E73D0	20 Fixed
040E73F0	20 Fixed
040E7410	10 Free
040E7420	20 Fixed
040E7440	30 Fixed
040E7470	20 Fixed
040E7490	20 Fixed
040E74B0	10 Free
040E74C0	20 Fixed
040E74E0	30 Fixed
040E7510	20 Fixed
040E7530	20 Fixed
040E7550	20 Free
040E7570	20 Fixed
040E7590	20 Fixed
040E75B0	20 Free
040E75D0	30 Fixed
040E7600	30 Fixed
040E7630	20 Fixed
040E7650	10 Free
040E7660	20 Fixed
040E7680	30 Fixed
040E76B0	20 Fixed
040E76D0	20 Fixed
040E76F0	20 Free
040E7710	30 Fixed
040E7740	30 Fixed
040E7770	20 Fixed
040E7790	20 Fixed
040E77B0	20 Free
040E77D0	30 Fixed
040E7800	30 Fixed
040E7830	20 Free
040E7850	20 Fixed
040E7870	20 Fixed
040E7890	10 Free

040E78A0 20 Fixed
040E78C0 30 Fixed
040E78F0 20 Fixed
040E7910 20 Fixed
040E7930 10 Free
040E7940 20 Fixed
040E7960 30 Fixed
040E7990 20 Fixed
040E79B0 20 Fixed
040E79D0 20 Free
040E79F0 30 Fixed
040E7A20 30 Fixed
040E7A50 20 Fixed
040E7A70 20 Fixed
040E7A90 10 Free
040E7AA0 20 Fixed
040E7AC0 30 Fixed
040E7AF0 20 Fixed
040E7B10 20 Fixed
040E7B30 10 Free
040E7B40 20 Fixed
040E7B60 30 Fixed
040E7B90 30 Fixed
040E7BC0 30 Fixed
040E7BF0 20 Fixed
040E7C10 20 Free
040E7C30 20 Fixed
040E7C50 20 Free
040E7C70 20 Fixed
040E7C90 20 Fixed
040E7CB0 10 Free
040E7CC0 20 Fixed
040E7CE0 30 Fixed
040E7D10 20 Fixed
040E7D30 20 Fixed
040E7D50 10 Free
040E7D60 20 Fixed
040E7D80 30 Fixed
040E7DB0 20 Fixed
040E7DD0 20 Fixed
040E7DF0 20 Free
040E7E10 30 Fixed
040E7E40 30 Fixed
040E7E70 20 Fixed
040E7E90 20 Fixed
040E7EB0 10 Free
040E7EC0 20 Fixed
040E7EE0 30 Fixed
040E7F10 20 Fixed
040E7F30 20 Free
040E7F50 30 Fixed
040E7F80 1A0 Fixed
040E8120 20 Fixed
040E8140 20 Fixed
040E8160 10 Free
040E8170 20 Fixed
040E8190 30 Fixed
040E81C0 20 Fixed
040E81E0 20 Fixed
040E8200 10 Free
040E8210 20 Fixed
040E8230 30 Fixed
040E8260 100 Free
040E8360 220 Fixed
040E8580 30 Fixed
040E85B0 B0 Free
040E8660 50 Fixed
040E86B0 50 Fixed
040E8700 A0 Free
040E87A0 20 Fixed
040E87C0 20 Fixed
040E87E0 10 Free
040E87F0 20 Fixed
040E8810 30 Fixed
040E8840 50 Fixed
040E8890 20 Free
040E88B0 20 Fixed
040E88D0 20 Fixed
040E88F0 10 Free
040E8900 20 Fixed
040E8920 30 Fixed
040E8950 20 Fixed
040E8970 20 Fixed
040E8990 20 Free
040E89B0 30 Fixed
040E89E0 30 Fixed
040E8A10 9E0 Fixed
040E93F0 20 Fixed
040E9410 60 Fixed
040E9470 20 Free
040E9490 20 Fixed
040E94B0 20 Fixed
040E94D0 10 Free
040E94E0 20 Fixed
040E9500 30 Fixed
040E9530 20 Fixed
040E9550 20 Fixed
040E9570 20 Free
040E9590 30 Fixed
040E95C0 30 Fixed
040E95F0 20 Fixed
040E9610 20 Fixed
040E9630 20 Free
040E9650 30 Fixed
040E9680 30 Fixed
040E96B0 20 Fixed
040E96D0 20 Fixed
040E96F0 20 Free
040E9710 30 Fixed
040E9740 30 Fixed
040E9770 60 Free
040E97D0 20 Fixed

040E97F0 20 Fixed
040E9810 20 Free
040E9830 30 Fixed
040E9860 10 Free
040E9870 20 Fixed
040E9890 30 Fixed
040E98C0 110 Free
040E99D0 20 Fixed
040E99F0 20 Fixed
040E9A10 20 Free
040E9A30 30 Fixed
040E9A60 30 Fixed
040E9A90 1B0 Free
040E9C40 20 Fixed
040E9C60 20 Fixed
040E9C80 10 Free
040E9C90 20 Fixed
040E9CB0 30 Fixed
040E9CE0 40 Free
040E9D20 20 Fixed
040E9D40 20 Fixed
040E9D60 20 Fixed
040E9D80 20 Fixed
040E9DA0 20 Fixed
040E9DC0 20 Fixed
040E9DE0 20 Fixed
040E9E00 20 Fixed
040E9E20 30 Fixed
040E9E50 20 Fixed
040E9E70 20 Fixed
040E9E90 10 Free
040E9EA0 20 Fixed
040E9EC0 30 Fixed
040E9EF0 20 Fixed
040E9F10 20 Fixed
040E9F30 20 Free
040E9F50 30 Fixed
040E9F80 30 Fixed
040E9FB0 20 Fixed
040E9FD0 20 Fixed
040E9FF0 2C0 Fixed
040EA2B0 230 Fixed
040EA4E0 230 Fixed
040EA710 230 Fixed
040EA940 230 Fixed
040EAB70 E0 Free
040EAC50 20 Fixed
040EAC70 20 Fixed
040EAC90 10 Free
040EACA0 20 Fixed
040EACC0 30 Fixed
040EACF0 20 Fixed
040EAD10 20 Fixed
040EAD30 10 Free
040EAD40 20 Fixed
040EAD60 30 Fixed
040EAD90 140 Free
040EAED0 20 Fixed
040EAEF0 20 Fixed
040EAF10 10 Free
040EAF20 20 Fixed
040EAF40 30 Fixed
040EAF70 20 Fixed
040EAF90 20 Fixed
040EAFB0 10 Free
040EAFC0 20 Fixed
040EAFE0 40 Free
040EB020 20 Fixed
040EB040 20 Fixed
040EB060 20 Free
040EB080 30 Fixed
040EB0B0 30 Fixed
040EB0E0 200 Free
040EB2E0 20 Fixed
040EB300 20 Fixed
040EB320 80 Free
040EB3A0 30 Fixed
040EB3D0 30 Fixed
040EB400 30 Fixed
040EB430 20 Fixed
040EB450 30 Fixed
040EB480 70 Fixed
040EB4F0 30 Fixed
040EB520 40 Fixed
040EB560 20 Fixed
040EB580 720 Fixed
040EBCA0 20 Fixed
040EBCC0 20 Fixed
040EBCE0 40 Free
040EBD20 30 Fixed
040EBD50 30 Fixed
040EBD80 20 Fixed
040EBDA0 100 Free
040EBEA0 20 Fixed
040EBEC0 20 Fixed
040EBEE0 60 Free
040EBF40 30 Fixed
040EBF70 30 Fixed
040EBFA0 30 Fixed
040EBFD0 780 Fixed
040EC750 250 Fixed
040EC9A0 60 Free
040ECA00 A0 Fixed
040ECAA0 30 Fixed
040ECAD0 120 Free
040ECBF0 60 Fixed
040ECC50 F0 Fixed
040ECD40 A0 Fixed
040ECDE0 A0 Fixed
040ECE80 20 Fixed
040ECEA0 20 Fixed
040ECEC0 20 Fixed

040ECEE0 20 Fixed
040ECF00 1A0 Free
040ED0A0 30 Fixed
040ED0D0 A0 Free
040ED170 40 Fixed
040ED1B0 A0 Fixed
040ED250 20 Fixed
040ED270 130 Free
040ED3A0 A0 Fixed
040ED440 30 Fixed
040ED470 20 Fixed
040ED490 20 Fixed
040ED4B0 30 Fixed
040ED4E0 20 Free
040ED500 20 Fixed
040ED520 20 Fixed
040ED540 20 Free
040ED560 30 Fixed
040ED590 30 Fixed
040ED5C0 20 Fixed
040ED5E0 20 Fixed
040ED600 20 Free
040ED620 30 Fixed
040ED650 30 Fixed
040ED680 A0 Fixed
040ED720 200 Free
040ED920 A0 Fixed
040ED9C0 A0 Free
040EDA60 30 Free
040EDA90 40 Free
040EDAD0 180 Free
040EDC50 20 Fixed
040EDC70 20 Free
040EDC90 20 Fixed
040EDCB0 20 Fixed
040EDCD0 A0 Free
040EDD70 30 Free
040EDDA0 40 Free
040EDDE0 220 Free

Processes and their Modules
Module File Attributes Legend
RO - Read Only
H - Hidden
S - System
A - Archive
C - Compressed

NK.EXE base address: C2000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP

filesys.exe base address: 04000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP
oemregistry.dll 02670000 13000 RO, H, S, C, XIP
relfsd.dll 02700000 7000 RO, H, S, C, XIP
fatfsd.dll 02710000 11000 RO, H, S, C, XIP
binfs.dll 03FB0000 8000 RO, H, S, XIP
flashdrv.dll 03FC0000 8000 RO, H, S, XIP
mspart.dll 03FD0000 8000 RO, H, S, C, XIP
fsdmgr.dll 03FE0000 11000 RO, H, S, XIP

device.exe base address: 06000000
==========
rsaenh.dll 018B0000 1A000 RO, H, S, C, RAM from ROM
btceif.dll 018D0000 18000 RO, S, C, RAM from ROM
oemnotify.dll 019A0000 A000 RO, H, S, XIP
battdrvr.dll 019B0000 9000 RO, H, S, XIP
frontlight.dll 019C0000 7000 RO, H, S, XIP
sdmemory.dll 019D0000 8000 RO, H, S, XIP
xscsdcard.dll 019E0000 8000 RO, H, S, XIP
sdbusdriver.dll 019F0000 D000 RO, H, S, XIP
coredll.dll 01F60000 85000 RO, H, S, XIP
ser2410.dll 021F0000 9000 RO, H, S, XIP
irsir.dll 02200000 7000 RO, H, S, XIP
pcmcia.dll 02210000 10000 RO, H, S, XIP
sc2410_usb_ser.dll 02220000 9000 RO, H, S, XIP
wavedev.dll 02230000 B000 RO, H, S, XIP
bthuniv.dll 024E0000 7000 RO, H, S, C, XIP
btd.dll 02520000 4C000 RO, H, S, C, XIP
crypt32.dll 02690000 42000 RO, H, S, C, XIP
ceddk.dll 026E0000 6000 RO, H, S, C, XIP
pm.dll 02730000 C000 RO, H, S, C, XIP
regenum.dll 02740000 5000 RO, H, S, C, XIP
wapdrv.dll 028B0000 1C000 RO, S, C, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
btdrt.dll 029A0000 F000 RO, H, S, C, XIP
msasn1.dll 029B0000 E000 RO, H, S, C, XIP
ndisuio.dll 029C0000 8000 RO, H, S, C, XIP
ndis.dll 029D0000 23000 RO, H, S, C, XIP
afd.dll 02A00000 16000 RO, H, S, C, XIP
spnego.dll 02A20000 C000 RO, H, S, C, XIP
ircomm.dll 02A30000 6000 RO, H, S, C, XIP
pptp.dll 02A40000 11000 RO, H, S, C, XIP
ethman.dll 02A60000 6000 RO, H, S, C, XIP
audevman.dll 02A70000 A000 RO, H, S, C, XIP
msim.dll 02EE0000 33000 RO, S, C, XIP
msnsspc.dll 02F60000 9000 RO, S, C, XIP
shutil.dll 02F80000 8000 RO, H, S, C, XIP
aygshell.dll 02FD0000 23000 RO, H, S, C, XIP
commctrl.dll 03040000 5B000 RO, H, S, C, XIP
netui.dll 030A0000 1A000 RO, H, S, C, XIP
bthlink.dll 03150000 8000 RO, H, S, C, XIP

redir.dll 03200000 24000 RO, H, S, C, XIP
unimodem.dll 03230000 E000 RO, H, S, C, XIP
tapi.dll 03240000 13000 RO, H, S, C, XIP
irdastk.dll 03260000 13000 RO, H, S, C, XIP
gsm610.acm 03280000 C000 RO, H, S, C, XIP
waveapi.dll 03290000 1B000 RO, H, S, C, XIP
softkb.dll 032B0000 9000 RO, H, S, C, XIP
netbios.dll 03750000 C000 RO, H, S, C, XIP
tcpstk.dll 03760000 58000 RO, H, S, C, XIP
ipv6hlp.dll 037C0000 B000 RO, H, S, C, XIP
tcpip6.dll 037D0000 54000 RO, H, S, C, XIP
dhcp.dll 03830000 A000 RO, H, S, C, XIP
eap.dll 03860000 8000 RO, H, S, C, XIP
eapol.dll 03870000 9000 RO, H, S, C, XIP
wzcsvc.dll 03890000 F000 RO, H, S, C, XIP
ntlmssp.dll 038A0000 D000 RO, H, S, C, XIP
schannel.dll 038B0000 1E000 RO, H, S, C, XIP
secur32.dll 038D0000 7000 RO, H, S, C, XIP
ssllsp.dll 038E0000 A000 RO, H, S, C, XIP
nspm.dll 038F0000 6000 RO, H, S, C, XIP
wspm.dll 03900000 6000 RO, H, S, C, XIP
ws2instl.dll 03910000 5000 RO, H, S, C, XIP
ws2.dll 03920000 C000 RO, H, S, C, XIP
winsock.dll 03930000 5000 RO, H, S, C, XIP
iphlpapi.dll 03940000 10000 RO, H, S, C, XIP
cxport.dll 03950000 6000 RO, H, S, C, XIP
asyncmac.dll 03960000 7000 RO, H, S, C, XIP
ppp.dll 03970000 1D000 RO, H, S, C, XIP
tshres.dll 7FFE0000 17000 RO, S, C, XIP

gwes.exe base address: 08000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP
keybddr.dll 02630000 8000 RO, H, S, XIP
touch.dll 02640000 B000 RO, H, S, XIP
ddi.dll 02650000 1C000 RO, H, S, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
shutil.dll 02F80000 8000 RO, H, S, C, XIP
aygshell.dll 02FD0000 23000 RO, H, S, C, XIP
commctrl.dll 03040000 5B000 RO, H, S, C, XIP
tshres.dll 7FFE0000 17000 RO, S, C, XIP

connmgr.exe base address: 0A000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP
configmanager.dll 02860000 10000 RO, S, C, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
cspproxy.dll 03160000 8000 RO, S, C, XIP
cspnet.dll 03170000 9000 RO, S, C, XIP
cspras.dll 03180000 E000 RO, S, C, XIP
connplan.dll 03190000 A000 RO, S, C, XIP
cellcore.dll 031B0000 8000 RO, S, C, XIP
wininet.dll 03600000 78000 RO, H, S, C, XIP
shlwapi.dll 03690000 13000 RO, H, S, C, XIP
ssllsp.dll 038E0000 A000 RO, H, S, C, XIP
wspm.dll 03900000 6000 RO, H, S, C, XIP
ws2.dll 03920000 C000 RO, H, S, C, XIP
iphlpapi.dll 03940000 10000 RO, H, S, C, XIP

shell32.exe base address: 0C000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP
toolhelp.dll 026F0000 5000 RO, H, S, C, XIP
wap.dll 028A0000 6000 RO, S, C, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
btdrt.dll 029A0000 F000 RO, H, S, C, XIP
calstore.dll 02CF0000 1A000 RO, S, C, XIP
pimutil.dll 02E10000 28000 RO, S, C, XIP
chngtrk.dll 02E40000 A000 RO, S, C, XIP
outres.dll 02E60000 38000 RO, S, C, XIP
shutil.dll 02F80000 8000 RO, H, S, C, XIP
aygshell.dll 02FD0000 23000 RO, H, S, C, XIP
ceshell.dll 03000000 37000 RO, H, S, C, XIP
commctrl.dll 03040000 5B000 RO, H, S, C, XIP
cellcore.dll 031B0000 8000 RO, S, C, XIP
webview.dll 033F0000 B7000 RO, H, S, C, XIP
browser.dll 034B0000 2E000 RO, H, S, C, XIP
imaging.dll 035B0000 49000 RO, H, S, C, XIP
shlwapi.dll 03690000 13000 RO, H, S, C, XIP
ssllsp.dll 038E0000 A000 RO, H, S, C, XIP
wspm.dll 03900000 6000 RO, H, S, C, XIP
ws2.dll 03920000 C000 RO, H, S, C, XIP
winsock.dll 03930000 5000 RO, H, S, C, XIP
iphlpapi.dll 03940000 10000 RO, H, S, C, XIP
pimapi.dll 03EA0000 7000 RO, S, C, XIP
tshres.dll 7FFE0000 17000 RO, S, C, XIP

services.exe base address: 0E000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
btdrt.dll 029A0000 F000 RO, H, S, C, XIP
obexsrvr.dll 031C0000 A000 RO, H, S, C, XIP
ssllsp.dll 038E0000 A000 RO, H, S, C, XIP
wspm.dll 03900000 6000 RO, H, S, C, XIP
ws2.dll 03920000 C000 RO, H, S, C, XIP
iphlpapi.dll 03940000 10000 RO, H, S, C, XIP

fexplore.exe base address: 10000000
==========

coredll.dll 01F60000 85000 RO, H, S, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
compime.dll 02ED0000 E000 RO, H, S, C, XIP
shutil.dll 02F80000 8000 RO, H, S, C, XIP
aygshell.dll 02FD0000 23000 RO, H, S, C, XIP
ceshell.dll 03000000 37000 RO, H, S, C, XIP
commctrl.dll 03040000 5B000 RO, H, S, C, XIP
cellcore.dll 031B0000 8000 RO, S, C, XIP
webview.dll 033F0000 B7000 RO, H, S, C, XIP
browser.dll 034B0000 2E000 RO, H, S, C, XIP
shlwapi.dll 03690000 13000 RO, H, S, C, XIP
ws2.dll 03920000 C000 RO, H, S, C, XIP
iphlpapi.dll 03940000 10000 RO, H, S, C, XIP
tshres.dll 7FFE0000 17000 RO, S, C, XIP

BTTrayCE.exe base address: 12000000
==========
btrez.dll 01800000 46000 RO, S, C, RAM from ROM
btchooserlib.dll 01850000 27000 RO, S, C, RAM from ROM
wbtapice.dll 01880000 C000 RO, S, C, RAM from ROM
mfcce300.dll 01C40000 6A000 RO, S, C, XIP
coredll.dll 01F60000 85000 RO, H, S, XIP
toolhelp.dll 026F0000 5000 RO, H, S, C, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
doclist.dll 02C20000 C000 RO, S, C, XIP
note_prj.dll 02D40000 13000 RO, S, C, XIP
shutil.dll 02F80000 8000 RO, H, S, C, XIP
aygshell.dll 02FD0000 23000 RO, H, S, C, XIP
ceshell.dll 03000000 37000 RO, H, S, C, XIP
commctrl.dll 03040000 5B000 RO, H, S, C, XIP
tshres.dll 7FFE0000 17000 RO, S, C, XIP

poutlook.exe base address: 14000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
pimutil.dll 02E10000 28000 RO, S, C, XIP
outres.dll 02E60000 38000 RO, S, C, XIP
shutil.dll 02F80000 8000 RO, H, S, C, XIP
aygshell.dll 02FD0000 23000 RO, H, S, C, XIP
commctrl.dll 03040000 5B000 RO, H, S, C, XIP
tshres.dll 7FFE0000 17000 RO, S, C, XIP

DUMPMEM.EXE base address: 16000000
==========
coredll.dll 01F60000 85000 RO, H, S, XIP
toolhelp.dll 026F0000 5000 RO, H, S, C, XIP
ossvcs.dll 028D0000 28000 RO, S, C, XIP
oleaut32.dll 02940000 30000 RO, H, S, C, XIP
ole32.dll 02970000 2F000 RO, H, S, C, XIP
compime.dll 02ED0000 E000 RO, H, S, C, XIP
shutil.dll 02F80000 8000 RO, H, S, C, XIP
aygshell.dll 02FD0000 23000 RO, H, S, C, XIP
commctrl.dll 03040000 5B000 RO, H, S, C, XIP
tshres.dll 7FFE0000 17000 RO, S, C, XIP

Appendix F – HERO Paper

Handheld Evidence Recovery Operator

Dr Kamal Jabbour, Sunderam Sankaran, Thomas N.J. Vestal
{jabbour, ssankara, tnvestal}@syr.edu
Syracuse University, Syracuse, NY

Abstract

Handheld devices present a class of computing platforms that provide its users mobility, connectivity, accessibility and organization. They also present a class of devices that are increasingly found during the course of legal investigations. Evidence acquisition tools for handheld devices are few, in contrast to the widespread use of these devices. There is a lack of clear understanding of the value of data contained in the devices and a standardized analysis process. In this paper, we discuss the general subject of forensic value of handheld devices by classifying the data they contain and data acquisition techniques. We introduce a prototype of an evidence acquisition tool, HERO, which uses Secure Digital storage cards for acquisition of data from Personal Digital Assistants running the Pocket PC operating system.

1. Introduction

A handheld device is characterized by an embedded operating system, a very small form factor, limited user input capability, limited storage, limited power and portability. The operating system (OS) is stored in a read only memory (ROM). Newer devices use flash drives for ROM to allow easy upgrades and updates to the OS. Some of the OS files are stored in a compressed format. The compressed files are loaded into main memory and decompressed before use. There also exist regions called Execute in Place (XIP). The files stored in these regions are capable of executing directly from ROM without loading into Random Access Memory (RAM). Handheld devices running the Pocket PC OS, for example, usually support XIP regions. The XIP regions allow OS modules to execute from the ROM thereby providing the user additional RAM for other applications. The RAM in mobile devices is unique since it serves as traditional memory for running applications and as storage for user data. This is in contrast to the PC where magnetic hard disk drives are used for storage. The amount of space allocated for storage and main memory is adjustable in many devices using a user interface. The file system used internally may not be apparent to a user. The user is presented with a unified view of the ROM, RAM and any other external storage. An example of this is the PIM information on a Pocket PC which is stored internally in the CE database format while the user views the data through an interface that hides the existence of the database. The user interface in handheld devices is very minimal and depends on the primary application of the device. A Personal Digital Assistant (PDA) has a more comprehensive interface than a cellular phone. Many of these devices have provisions for external storage cards. Handheld devices are powered by batteries and hence have limited power. The devices, operating system and applications are optimized for low power consumption. The characteristics of handheld devices discussed so far present a need for tools and techniques designed for these devices to conduct forensic investigations. The need is increasing as these

devices become more powerful, integrated and easy to use. Handheld devices provide a wealth of information and can be significant in a forensic investigation. We present a classification of data into common types found in handheld devices along with acquisition and analysis techniques for the data. The final section discusses the Handheld Evidence Recovery Operator (HERO) developed for the Pocket PC in detail.

2. Handheld Forensics

This paper deals with a set of digital sources collectively called handheld devices. Cell phones, personal digital assistants (PDA), smartphone, blackberries, smartwatches, Voice over Internet Protocol (VoIP) phones present a sampling of handheld devices found in the consumer market. The popular operating systems found on these devices are the Pocket PC (based on Windows CE), Palm OS, Symbian and sometimes Linux. Each of these devices has unique features and characteristics. They also have some commonalities.

2.1 Data Classification

Handheld devices could effectively provide the following classes of information.

a. Personal: Information regarding personal contact information such as name, address and home telephone number is usually available on the device. This is valuable in investigations where the suspect who possesses the device is unwilling to talk or the device is found in a situation as part of a search. Information can be obtained from data such as contact lists which contain names, addresses and phone numbers of people, to-do lists that contain tasks to be completed, appointments, and notes taken at a meeting, voice memos, bank accounts and purchase orders. The information from the above data is valuable in understanding the suspect's past, current and future activities, and associations which may provide probable cause evidence to issue a comprehensive search warrant.

b. Archived: Information that is present on the device but is not evident is classified as archived. This type of information is gathered from data stored in files and organized in directories. Files can contain images, audio, video, executables, contacts, notes, letters and more. The information from these sources could provide evidence against a suspect in a criminal proceeding or information that could lead to further investigation.

c. Residual: This kind of information results from a number of activities, either by the user or the device. User activities include modification, deletion, creation and access. Some devices maintain logs of user activities. Creating or modifying data can result in the activity getting logged. This provides a means for non-repudiation in many cases. The log data could be modified to alter contents but such an act usually leaves the file modification date and time changed. The log file can be deleted. Most devices do not delete data when requested to do so, instead the entry for the file containing the data is removed from the allocation table and the space is marked as available for re-use. The device activities such as the last time a reboot was performed, new programs installed or uninstalled, location of the device where it was last used, the last user to log on to the

device, the files that were uploaded/downloaded to/from another device, last incoming/outgoing phone call, and the last access point that was discovered. This logged data is useful in narrowing the activities performed by the user of the device before it was discovered.

d. Hidden: This type of information is found in slack spaces of storage drives and information hidden inside other files. The existence of this information is itself evidence of illegal activity in many cases. Image, audio and video files can contain data hidden inside using steganographic methods.

e. Embedded: Data embedded in to the devices include PINs, manufacturer ID, network ID, make, model and serial number of the device. This information is helpful to obtain logs for the device from the network service provider and manuals from the manufacturer.

The data types described here are the target of data acquisition. All of the data types may not be present in a given device. Knowledge of device type and manufacturer documentation provides the investigator useful information before proceeding to the acquisition phase.

2.2 Data Acquisition Techniques

There are two classes of techniques for data acquisition from mobile devices. They are Physical and Peripheral. These are discussed here.

2.2.1 Physical: Situations where data is not recoverable using software, hardware devices are used for acquisition. Data stored on read only memory can be recovered using hardware interfaces that are built into the device. The manufacturer sometimes has interfaces for debugging and troubleshooting devices such as JTAG interfaces. These can be used to probe and read data that are otherwise inaccessible. Hardware readers for specific parts of the device also exist. This acquisition technique is not covered in this paper.

2.2.2 Peripheral: The case when the investigator has unrestricted access to the device, software tools can help acquire evidence. Handheld devices do not have removable primary storage. The OS in most cases resides in ROM and is constantly running. Traditional tools like boot disks cannot be used in this case. The OS then needs to participate in the acquisition process. The acquisition tool has several channels to get injected and executed on the device. Handheld devices have a data port for synchronization with a desktop computer. This feature is common to most devices in the market. The OS vendors usually have a remote API set for execution on the desktop. The remote API issues commands through the synchronization software to the device. The device runs a synchronization client that relays the commands to the OS and returns the results. Several tools utilize this for acquisition.

Another possibility is to use secondary storage for acquisition. SD card and CF card slots are common in many of the devices. The acquisition tool is stored in the card and invoked when the card is inserted into the device. The acquired data is stored on the card for further analysis. SD cards with 1 gigabyte capacity are available and are sufficient for devices available today.

3. Handheld Evidence Recovery Operator (HERO)

The Handheld Evidence Recovery Operator (HERO) developed at the Air Force Research Laboratory is a set of tools targeted at handheld devices with the ARM processor running Pocket PC operating system.

3.1 Pocket PC: The Pocket PC operating system is a derivative of Windows CE. Each version of the OS reflects a different version of the base Windows CE version. The popular versions are the Pocket PC 2002 phone edition and the Pocket PC 2003. The OS is run primarily on ARM core based processors such as the Intel XScale processor. The user interface is different from the Windows Desktop environment but gives a familiar set of system file names and icons. The user views the data in terms of files and folders similar to the Windows desktop. Internally the OS does not have the Current Folder or drive concept of the PC. Instead all folders are referenced relative to the root ("\"). For example, the folder C:\Windows\System on the PC is \Windows\System on the Pocket PC. The files in ROM and RAM are presented in a unified view and the user cannot differentiate between files in the ROM and those in RAM. Peripheral devices such as SD cards and CF cards are presented as folders in the root of the device. A SD card in a HP iPAQ with a 233 MHz Intel XScale processor running Pocket PC 2003, for example, will show as "Storage Card" in the root.

The Application Programming Interface (API) for the Pocket PC is a subset of the API for Windows on the PC. This gives traditional Windows programmers a smooth transition into the Pocket PC arena. From a forensic tool development standpoint, the API provides easy access to the data. The Pocket PC development environment provides a fairly complete set of Windows API, a C library and an MFC library as dynamic link libraries (DLL).

The Pocket PC also has a feature called the "Autorun" which allows the device to automatically execute a program stored on a peripheral storage card. The executable is stored in a specific folder with a specific name (autorun.exe for example) on the SD card. The name of the folder depends on the processor type. This feature is optional and may not be implemented on all devices.

The next section describes the set of tools developed for acquisition and analysis of data from Pocket PC based devices.

3.2 Acquisition: HERO has a set of five tools to acquire data from a given device. They reside on a SD card and execute in the main memory of the device. Most SD cards have a write lock which helps in preventing accidental writes to the card after the acquisition is complete. The executables are no more than 10 kilo bytes in size. The tools are capable of extracting the ROM, object store, databases, files and the registry.

The SD card is prepared with appropriate steps to sanitize the sectors. The card is formatted to support the FAT32 file system. The Pocket PC supports the FAT file system in addition to the Object Store.

The tools are stored on the SD card at a convenient location. Depending on the situation, the autorun feature can be used or the programs can be executed manually by using the user interface. Each tool determines the location and name of the SD card since the Pocket PC does not support the current directory feature. A progress bar indicates the state of the acquisition.

The files and directories on the device are copied as they are, maintaining relative positions, onto the SD card. The files copied include hidden files that may not be visible on the device. The tool that extracts the object store captures the files and directories in their compressed form. This does not include the files in ROM. The copy of the registry in RAM and the databases are present in the image. This is the bit image of the object store. The bit image is valuable when extracting deleted files and examining the free space. The database tool extracts the CE databases that store personal information. The extracted databases are stored in a CE database file. This file is in a format that is readable by another Pocket PC device with the same or similar characteristics as the original device. This is helpful when the device cannot be directly used for analysis.

The acquisition process can be automated by having an autorun program invoke each tool without user interruption. The autorun program can be set to execute when the SD card is inserted into the device. The advantage of using a SD card for acquisition is that it voids the need for equipment to be carried around. In addition, it serves the dual purpose of a field analysis tool for time-critical intelligence gathering.

3.3 Analysis: The evidence collected is stored on the SD card and the write lock is set. The analysis is done using a card reader and a PC with Windows XP Professional. The analysis software is installed on the PC. The installation is a self extracting executable that creates the appropriate folders, copies the files and creates registry keys. The program provides a graphical interface to navigate through the SD card. The program is aimed at assisting the investigator in the analysis process. There are several tools built in to help the analysis.

File Search: The file search utility searches for and lists files found based on the pattern specified. The pattern is a file name followed by its extension. The user can specify part of a file name to search for files with that pattern in the name. The option of searching in the current directory and sub-directories is available. The search function examines both the extension in the file name and the extension in the file header. The two are compared to determine if there is a discrepancy. Windows operating systems associate files with icons based on the extension in their name. This feature is helpful in finding files that are renamed to hide their original content. A separate list of files with extensions different from their headers is generated. Search filters such as date, time and size are available to narrow the search results. An indicator displays the progress of the search. The user can display the files in hex or ASCII format to examine the internals of the file. Image files are viewable in an image viewer in addition to the hex/ASCII display.

Keyword Search: The keyword search utility allows the user to search inside the files. The user specifies a keyword in hex or ASCII to search for within the files. This is sometimes helpful to understand the intended use of the file. For example, static strings within executables could reveal its purpose without the user having to execute the program.

Steganography detection: Steganography is the art of hiding information within messages. This utility allows the user to scan files for hidden information. The tools used are commonly available tools such as stegdetect. The utility allows the user to add new tools to the set.

Hash calculation: The user can calculate the MD5 hash of a string or file. This is useful to compare hash values of files to determine if they are modified. The MD5 implementation is derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm as specified in RFC 1321 by the IETF [2].

File recovery: The object store image obtained from the handheld device is a snapshot of the object store at the byte level. The Pocket PC OS does not completely delete a file. The OS assigns free blocks of storage as needed. This leaves fragments from previously deleted files inside the object store. The utility scans through the object store image for file headers and creates a file for each header found.

Logging: The user actions are logged to a file. The actions logged include log in time, searches performed, files opened, utilities used and logout time. The log is frequently written to disk. This is useful if the program is interrupted and has to be closed abruptly. The log maintains date and time of the actions along with the user that performed the actions.

The program does not allow the user to edit files and opens files in read only mode. A universal filtering scheme allows the user to display only files that are targeted. This is useful, for example, in cases where only image files are of interest. The program has a sign-in and sign-off scheme to log user activity.

3.4 Results

Preliminary tests conducted with the HERO have returned positive results. File integrity tests were conducted to determine if the tool modified the files on the device. The MD5 algorithm was used for file integrity checks. The MD5 hash of files was calculated on a device before running the tool. The hashes were stored on a PC for safe keeping. We ran the HERO tool on the device capturing the image and files. The MD5 hash of the files was gathered again and compared with the original set. The values matched and we concluded that the integrity of files was maintained by the tool. We performed the same test two more devices and the results were the same.

The object store image was used to recover deleted files. We placed a text file and an image file in the root folder of the device and ran the tool. A search for the file names revealed their existence in the object store image. We noted down the offset within the image where the files were found. We then deleted the files on the device using the user interface. We ran the tool again on the device. The object store image retained the files at the same offset even after deletion. We were able to recover the files by copying the bytes and pasting into a new file.

References

[1] “Steganography Detection with Stegdetect”, Neils Provos, URL: http://www.outguess.org/detection.php

[2] “The MD5 Message-Digest Algorithm”, R. Rivest, MIT Laboratory for Computer Science and RSA Data Security, Inc., IETF Network Working Group RFC 1321, April 1992, URL: http://www.ietf.org/rfc/rfc1321.txt